T.C.
MİLLÎ EĞİTİM BAKANLIĞI

MEGEP

(MESLEKİ EĞİTİM VE ÖĞRETİM SİSTEMİNİN
GÜÇLENDİRİLMESİ PROJESİ)

ENDÜSTRİYEL OTOMASYON TEKNOLOJİLERİ

SCADA İLE VERİTABANI

ANKARA, 2009

Milli Eğitim Bakanlığı tarafından geliştirilen modüller;

- Talim ve Terbiye Kurulu Başkanlığının 02.06.2006 tarih ve 269 sayılı Kararı ile onaylanan, Mesleki ve Teknik Eğitim Okul ve Kurumlarında kademeli olarak yaygınlaştırılan 42 alan ve 192 dala ait çerçeve öğretim programlarında amaçlanan mesleki yeterlikleri kazandırmaya yönelik geliştirilmiş öğretim materyalleridir (Ders Notlarıdır).
- Modüller, bireylere mesleki yeterlik kazandırmak ve bireysel öğrenmeye rehberlik etmek amacıyla öğrenme materyali olarak hazırlanmış, denenmek ve geliştirilmek üzere Mesleki ve Teknik Eğitim Okul ve Kurumlarında uygulanmaya başlanmıştır.
- Modüller teknolojik gelişmelere paralel olarak, amaçlanan yeterliği kazandırmak koşulu ile eğitim öğretim sırasında geliştirilebilir ve yapılması önerilen değişiklikler Bakanlıkta ilgili birime bildirilir.
- Örgün ve yaygın eğitim kurumları, işletmeler ve kendi kendine mesleki yeterlik kazanmak isteyen bireyler modüllere internet üzerinden ulaşılabilirler.
- Basılmış modüller, eğitim kurumlarında öğrencilere ücretsiz olarak dağıtılır.
- Modüller hiçbir şekilde ticari amaçla kullanılamaz ve ücret karşılığında satılamaz.

İÇİNDEKİLER

AÇIKLAMALAR ii
GİRİŞ 1
ÖĞRENME FAALİYETİ–1 3
1. VERİ TABANINA GİRİŞ 3
1.1. Veri Tabanının Tanımı 3
1.2. Ms Access Veri Tabanı 4
1.2.1. Yeni Bir Veri Tabanı Açmak 4
1.2.2. Tablo Oluşturma 5
1.2.3. Form Oluşturma 8
1.2.4. Sorgu Oluşturma 11
1.2.5. Rapor Oluşturma 13
1.2.6. İlişkinin Tanımı 16
1.2.7. Veri Tabanı Modelleri 19
1.3. ODBC 21
1.4. MS SQL Server 2000 21
1.4.1. Giriş 21
1.4.2. Veri Tabanı, Hesap ve Tablolar Oluşturma 25
1.5. WinCC Ortamını Tanıma 28
1.6. WinCC'de Verileri Görme 38
UYGULAMA FAALİYETİ 49
ÖLÇME VE DEĞERLENDİRME 51
ÖĞRENME FAALİYETİ–2 52
2. VB İLE VERİ TABANI 52
2.1. Data Kontrol Bileşeni 52
2.2. Recordset 55
2.3. ADO 60
2.4. OPC 65
2.5. Omron CX OPC Server 67
2.6. Kepware OPC 84
UYGULAMA FAALİYETİ 91
ÖLÇME VE DEĞERLENDİRME 92
MODÜL DEĞERLENDİRME 93
CEVAP ANAHTARLARI 94
KAYNAKÇA 95

AÇIKLAMALAR

KOD	**523EO0378**
ALAN	**Endüstriyel Otomasyon Teknolojileri**
DAL/MESLEK	**Endüstriyel Kontrol Teknisyenliği**
MODÜLÜN ADI	**Scada İle Veritabanı**
MODÜLÜN TANIMI	Bu modül scada ile veri tabanı uygulamalrına yönelik bilgi ve becerilerin verildiği öğrenme materyalidir.
SÜRE	40/32
ÖN KOŞUL	SCADA sistemi kurma modülünü almış olmak
YETERLİK	SCADA sisteminde veri tabanı oluşturmak
MODÜLÜN AMACI	**Genel Amaç** Sistem verilerini veri tabanına hatasız olarak kaydedebileceksiniz. **Amaçlar** **1.** MS Access'te ve WinCC'ye veri tabanını öğreneceksiniz. **2.** VB ile PLC'den alınan bir verinin MS Access tablosuna kaydedebileceksiniz.
EĞİTİM ÖĞRETİM ORTAMLARI VE DONANIMLARI	Bilgisayar, PLC, gerekli bağlantı kabloları ve veri tabanı programı
ÖLÇME VE DEĞERLENDİRME	Modül içinde yer alan her öğrenme faaliyetinden sonra verilen ölçme araçları ile kendinizi değerlendireceksiniz. Öğretmen modül sonunda ölçme aracı (çoktan seçmeli test, doğru-yanlış vb.) kullanarak modül uygulamaları ile kazandığınız bilgi ve becerileri ölçerek sizi değerlendirecektir.

GİRİŞ

Sevgili Öğrenci,

SCADA terimi “supervisory control and data acquisition” kelimelerinin ilk harfleri ile oluşturulan bir kısaltmadır. SCADA sistemleri büyük bir alana ait teknik konulardaki işlevini yerine getirmek amacındadır. Bu bağlamda çoğunlukla HMI (human-machine ınterface) veya MMI (man-machine ınterface) kısaltmaları ile yan yana kullanılır.

İşlevleri:

- İzleme işlevleri (olay ve alarm işleme)
- Kontrol işlevleri
- Veri toplama*
- Verilerin kaydı ve saklanması

Bu modülde veri tabanı kavramı üzerinde durulacaktır.

ÖĞRENME FAALİYETİ–1

AMAÇ

MS Access'te ve WinCC'ye veri tabanını öğreneceksiniz.

ARAŞTIRMA

- SQL server konusunda ön bilgi ediniz.

1. VERİ TABANINA GİRİŞ

1.1. Veri Tabanının Tanımı

İnsan yaşamını bir bilgi deryası kuşatmıştır. Günlük yaşamı döndüren tüm hizmet ve faaliyetler düzenli bir kayıt altında alınmalı ve en küçük bir aksamaya yol açmayacak şekilde yürütülmelidir. Kredi kartlarından okul kayıtlarına, vergi hesaplarından üretim rakamlarına muazzam bir çark işlemektedir. Geçmişe kıyasla hayatın ritminin oldukça hızlandığı günümüzde bilgisayar teknolojisi bu hıza ayak uydurabilecek kavramlar üretmiştir. Bunlardan bir tanesi veritabanlarıdır.

Bilgisayar terimi aslında yapılan işi ortaya koyan bir terimdir. Bilgi saymak işlemi eğer elde düzenli ve işlenmeye hazır veri topluluğu varsa yapılabilecek bir işlemdir. İşte bu veri topluluğu veri tabanı adını alır. Daha nizami bir tanım yapılacak olursa; sistematik erişim imkânı olan, yönetilebilir, güncellenebilir, taşınabilir, birbirleri arasında tanımlı ilişkiler bulunabilen bilgiler kümesidir. Bilgisayar ortamında bu bilgi kümelerini düzenli bir şekilde depolayan yazılımlara ise veri tabanı yazılımları adı verilir.

Veri tabanı tıpkı bir ofisteki dosya dolabına benzetilebilir. Dosya dolabı içinde evraklar bulunmaktadır. Dosya dolabından sorumlu bir ofis personelinin bulunduğu düşünülürse aşağıdaki şekilde bir yapıya karşılık gelir.

Veri tabanı :Dosya dolabı
Kayıtlar :Evraklar
Veri tabanı yönetim sistemi: Ofis personeli

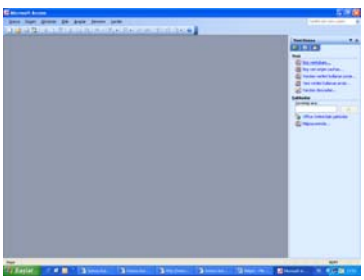
Veri bir kelime işlemcide olduğu gibi sadece metin tabanlı olmayıp kümeler hâlinde dosyalarda saklanır. Veri tabanında verileri saklayan dosyalara tablo denir. Müşteri tablosu, kitaplar tablosu gibi.

1.2. Ms Access Veri Tabanı

Microsoft firması tarafından üretilen ve Office paketi içinde sunulan Ms Access programı kullanıcılara orta seviye veri tabanı çözümleri sunan oldukça popüler bir yazılımdır. Access veri tabanı programını şekilde görüldüğü gibi başlatılır.

Şekil 1.1: 'Microsoft access'i başlatma

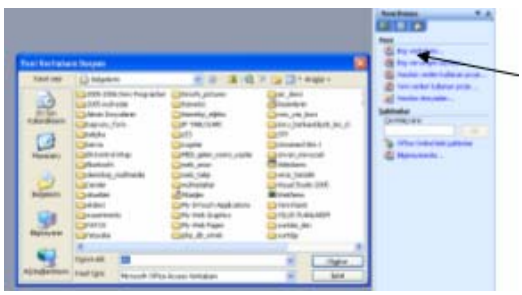
Access programı standart Ms Office arayüzüne sahiptir.



Şekil 1.2: Access'in arayüzü

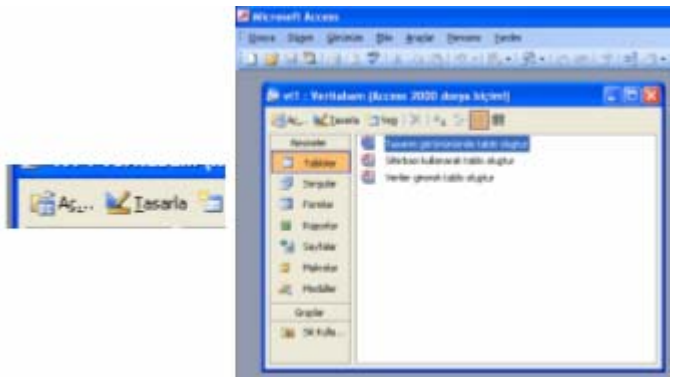
1.2.1. Yeni Bir Veri Tabanı Açmak

Programın arayüzünde yeni bir veri tabanı açmak için sağ taraftaki menüden "boş veri tabanı" seçeneği tıklanır ve ekrana gelen standart Windows diyalog penceresine isim ve kayıt yeri belirtilerek işlem gerçekleştirilir.



Şekil 1.3: Yeni veri tabanı oluşturma

vt1.mdb olarak veri tabanı oluşturulduğunda ekrana vt1'e ait kontrol penceresi gelir.



Şekil 1.4: Veri tabanını açma

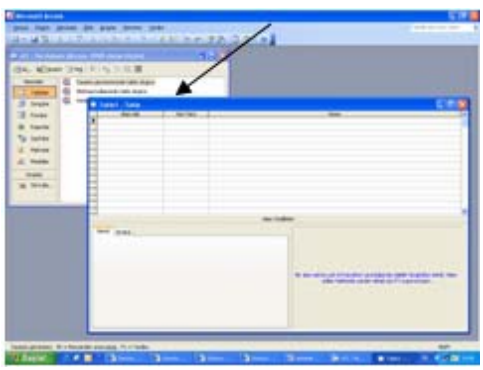
Kontrol penceresi kullanılarak tablo, sorgu, form gibi bileşenler oluşturulabilir.

1.2.2. Tablo Oluşturma

Kontrol penceresinin sol tarafından tablo seçildiğinde 3 yöntemle tablo oluşturulabilir:

- Tasarım görünümüyle
- Sihirbaz yardımı ile
- Doğrudan veri girilerek

Standart tablolar oluşturmak için sihirbaz kullanmak en kolay yoldur. Ama tablo tasarımında detayların önem kazandığı durumlarda genellikle tasarım görünümü ile tablolar oluşturmak gerekmektedir.



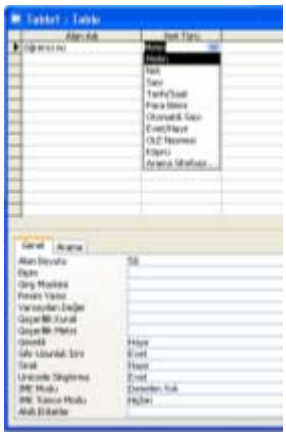
Tasarım görünümüyle tablo oluşturmak için çift tıklamak ya da tasarla seçeneğini tıklamak yeterlidir. Bu şekilde tasarım penceresi kullanıcı ekranına gelir.

Şekil 1.5: Tablo oluşturma

Tablo tasarım penceresinin üst kısmında alan adı, veri türü ve tanım sütunları bulunur. Bu sütunlara tablo tasarımında gerekli olan alanlar ve veri türleri yazılır. Veri türü açılan kutu şeklindedir. Alan özelliğine göre açılan kutu içindeki türler seçildiğinde bu türe ilişkin özellikler alt kısımda listelenir. Böylelikle her alana ait veri türü ve o türe ait özellikler tek tek ayarlanarak tasarım gerçekleştirilir. Veri türleri, tabloya ait alanlarına kaydedilecek verilerin türleri bu kısımda belirtilir.

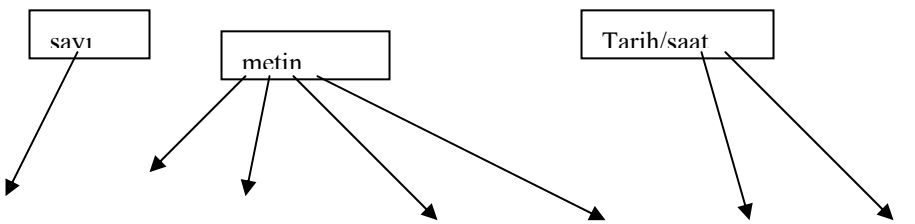
Metin	Metin veya metin ve sayı bileşimleri, örneğin adresler için veya hesaplama gerektirmeyen sayılar, örneğin telefon numaraları, bölüm numaraları veya posta kodları için kullanılır. En fazla 255 karakter. FieldSize özelliği girilebilen en fazla karakter sayısını denetler.
Kısa Not	Uzun metinler ve sayılar, örneğin notlar ve açıklamalar için kullanılır. En fazla 65.536 karakterdir.
Sayı	Para hesaplamaları dışında (para birimi türü) matematiksel hesaplamalara eklenen veriler için kullanılır. 1, 2, 4 veya 8 bayt; yineleme kimliği (GUID) için 16 bayt. FieldSize özelliği belirli bir sayı türü tanımlar.
Tarih/Saat	Tarih ve saat için kullanılır. 8 bayt.
Para Birimi	Para birimi değerlerinde ve hesaplama sırasında yuvarlamayı önlemek için kullanılır. 8 bayt.
Otomatik Sayı	Bir kayıt eklendiğinde otomatik olarak eklenen benzersiz sıralı (1'den başlayarak) veya rastgele numaralar için kullanılır. 4 bayt; yineleme kimliği (GUID) için 16 bayt.
Evet/Hayır	Yalnızca olası iki değerden biri olabilen veriler için kullanılır, örneğin Evet/Hayır, Doğru/Yanlış, Açık/Kapalı. Boş (Null) (**Boş**: Alana girebileceğiniz veya ifadelerde ve sorgularda eksik veya bilinmeyen verileri belirtmek için kullanabileceğiniz bir değer. Visual Basic'te, **Null** anahtar sözcüğü **Boş** değeri belirtir. Birincil anahtar alanları gibi bazı alanlar **Boş** değerler içeremez.) değerlere izin verilmez. 1 bit yer kaplar.
OLE Nesnesi	OLE (Programlar arasında bilgi paylaşımında bulunmak için kullanabileceğiniz bir program-tümleme teknolojisi. Tüm Office programları OLE'yi desteklediğinden bağlantılı ve katıştırılmış nesneler yoluyla bilgi paylaşımı yapabilirsiniz.) protokolü kullanılarak oluşturulmuş OLE nesneleri (örneğin Microsoft Word belgeleri, Microsoft Excel elektronik tabloları, resimler, sesler veya diğer ikili veriler) için kullanılır.
Köprüler	Köprüler (Bir dosyaya, dosya daki bir konuma, World Wide Web üzerindeki bir Web sayfasına veya bir intranet üzerindeki Web sayfasına gitmek için tıklattığınız renkli ve altçizgili metin veya grafik) için kullanılır.
OLE Nesnesi	OLE (Programlar arasında bilgi paylaşımında bulunmak için kullanabileceğiniz bir program-tümleme teknolojisi. Tüm Office programları OLE'yi desteklediğinden bağlantılı ve katıştırılmış nesneler yoluyla bilgi paylaşımı yapabilirsiniz.) protokolü kullanılarak oluşturulmuş OLE nesneleri (örneğin Microsoft Word belgeleri, Microsoft Excel elektronik tabloları, resimler, sesler veya diğer ikili veriler) için kullanılır. En çok 1 gigabayt (disk alanıyla sınırlı).
Arama Sihirbazı	Başka bir tablodan veya açılan kutu kullanan bir değerler listesinden değer seçmenizi sağlayan bir alan oluşturur. Veri türü listesinde bu seçenek seçildiğinde bu değeri tanımlayacak bir sihirbaz başlatılır.

Tablo 1.1: Veri türleri

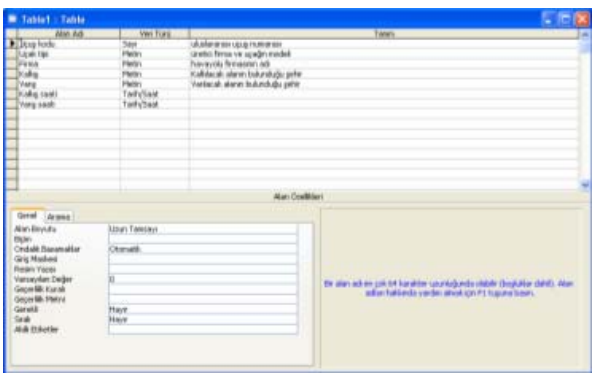


Şekil 1.6: Tablo alanlarının doldurulması

ÖRNEK 1: Havalimanı.mdb isimli veri tabanı açarak aşağıdaki gibi bir tablo (uçuş_bilgi) oluşturma



Uçuş kodu	Uçak tipi	Firma	Kalkış	Varış	Kalkış saati	Varış saati
52	Airbus A310	THY	İstanbul	Budapeşte	14:30	16:10
29	Boeing 747	Pegasus	İstanbul	Amsterdam	17:00	19:30



Şekil 1.7: Tablo oluşturma

Veri türlerine ait en yaygın kullanılan özellikler şunlardır:

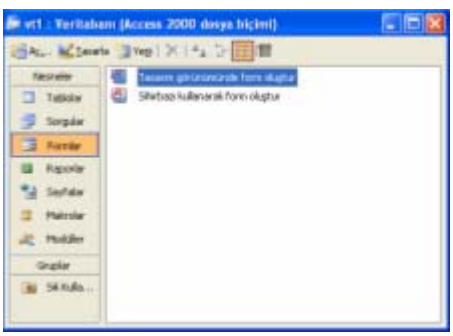
Giriş maskesi	Hücreye girilecek verinin biçimi tanımlanır.
Geçerlilik kuralı	Hücreye girilecek değerleri sınırlar.
Geçerlilik metni	Geçerlilik kuralına uymayan durumlarda görünecek uyarı mesajıdır.
Varsayılan değer	Hücreye bilgi girişi esnasında otomatik olarak gelmesi istenen değerdir.
Alan boyutu	Hücrenin barındırabileceği boyuttur.

Tablo 1.2: Veri türleri

1.2.3. Form Oluşturma

Access programı, veri tabanı içine kullanıcı arayüzü oluşturulmasına imkân tanır. Tablo yapısı meydana getirildikten sonra, kullanıcıların veri girişi yapabilmeleri, sorgu ve rapor işlemlerini ekranda düzenli bir şekilde görüp yorumlayabilmelerine imkân tanıyan Windows standart pencere yapısına uygun form pencereleri oluşturulabilir. Böylelikle herhangi ikinci bir yazılım ya da programlama diline ihtiyaç duyulmadan etkin uygulamalar geliştirilebilinir.

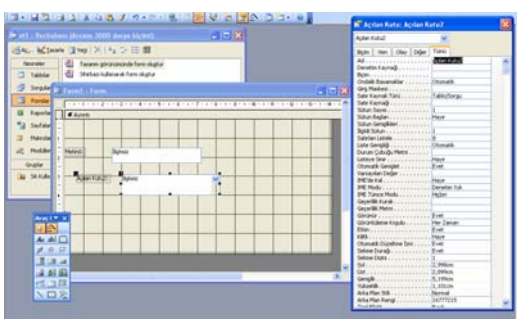
Tablo oluşturmadaki benzer adımlar form içinde geçerlidir. Kontrol penceresinde formlar seçeneği tıklandığında tasarım penceresi ve sihirbaz olarak iki yöntemle form oluşturulabileceği görülebilir.



Şekil 1.8: Form oluşturma

Eğer yaygın ve özel ayarlar gerektirmeyen bir form tasarlanacak ise sihirbaz yöntemi oldukça hızlı ve basit bir yoldur.

Tasarım görünümü ise önceden planlanmış ekran görüntüsüne karar verilmiş formlar oluşturmaya izin verir.



Şekil 1.9: Tasarımla form oluştruma

Yukarıdaki ekran görüntüsünden de anlaşılacağı üzere, açılan form penceresine araç kutusunda bulunan Windows nesneleri yerleştirilerek tasarımın görsel bölümü meydana getirilir. Görsel programlama (visual basic, delphi vb.) dillerine aşina olan programcılar için oldukça kolay bir aşamadır.

Form üzerine yerleştirilen her araç için kendine ait özellikler penceresinden gerekli ayarlamalar yapılabilir. Araçların oldukça detaylı ve programcıya esneklik tanıyan özellikleri bulunmaktadır (Eğer özellikler penceresi kapalı ise ilgili araç tıklanarak sağ tuş menüsünden özellikler seçildiğinde bu pencere görülebilir.).

Özellikler penceresi dikkatli incelendiğinde 5 adet sekmenin var olduğu görülebilir. Bunlar biçim-veri-olay-diğer-tümü şeklinde isimlendirilmiştir.

Biçim sekmesi	Bu kısımdaki ayarlar form üzerine yerleştirilen araca ait biçimsel ve görsel seçenekleri barındırır.
Veri sekmesi	Aracın eğer varsa tablolar ve alanlarla ilişkileri bu kısımda belirtilir.
Olay sekmesi	Görsel programcıların yabancı olmadıkları (event-driven) denilen olay-tabanlı çalıştırma ilişkisi burada yapılır. Windows programlarında çift-tıklama, tuşa basma gibi komutlar verilerek işlemlerin yapılmasına benzer şekilde, programcı geliştirdiği bir access projesine formlar tasarlayarak bunlara işlevsellik kazandırabilir.
Diğer sekmesi	Yukarıdaki üç kategorinin dışında kalan ayarlar buradadır.
Tümü sekmesi	Yukarıdaki dört sekmede bulunan özelliklerinin tamamı bu sekmede birlikte görülebilir.

Tablo 1.3: Form özellikler penceresi

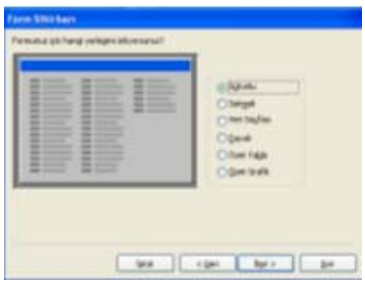
ÖRNEK 2:

Daha önceden hazırlanan havalimanı.mdb veri tabanı uygulamasındaki uçuş_bilgi tablosuna veri girişi amaçlı bir formu sihirbaz yardımı ile oluşturma

Öncelikle kontrol penceresinde sağ menüden form seçilir ve “sihirbaz kullanarak form oluştur ” seçeneği çift tıklanır.

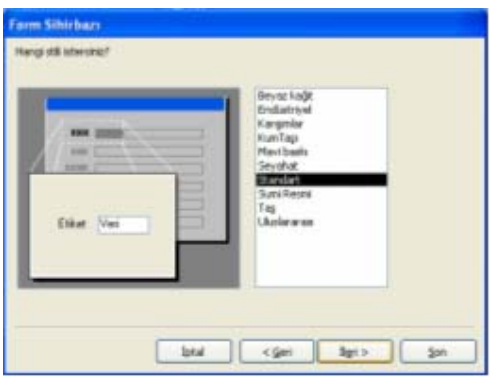
Şekil 1.10: Tasarım sihirbazı

Ekrana yukarıdaki sihirbazın ilk adımı gelir. Bu aşamada uçuş_bilgi tablosunda bulunan alanlardan form üzerinde kullanılacak olanlar sağ tarafa aktarılarak seçilir.



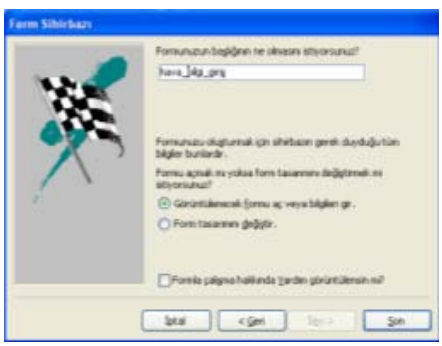
Şekil 1.11: Tasarım sihirbazı

İlgili alanların seçiminden sonra sihirbaz form düzeninin nasıl olacağını sorar.



Şekil 1.12: Tasarım sihirbazı

Bir sonraki aşamada form görünümüne karar verilir.



Şekil 1.13: Tasarım sihirbazı

Son aşamada forma bir isim verilir. Son butonu tıklandıktan sonra hazırlanan form ekranı aşağıdaki gibi görülür. Sihirbaz yardımıyla hazırlanan formlar üzerinde değişikliklere ihtiyaç olduğunda, formun tasarım görünümüne girilerek ayarlamalar yapılabilir.

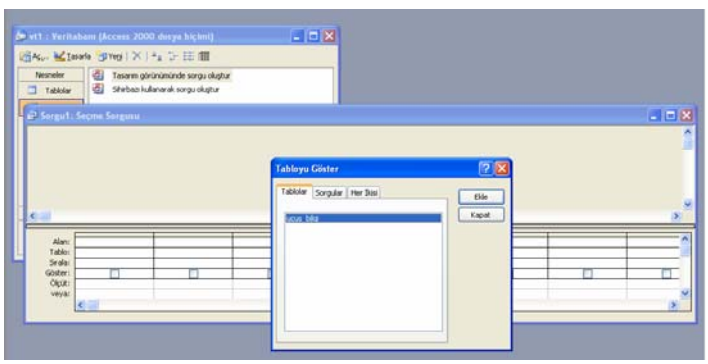
Şekil 1.14: Tasarımı test etme

1.2.4. Sorgu Oluşturma

Veri tabanı organizasyonlarında hedeflenen amaç binlerce veri yığını içinden istenen şartlara ve aralıklara denk düşen verilerin çekilebilmesidir. Bu şart ve aralıkları ifade edebilmek ancak bir sorgulama dili kullanılarak olur. Bilgisayar dünyasında en yaygın sorgulama dili olarak SQL (structural query language) görülmektedir. Türkçeye "yapısal sorgulama dili" olarak çevrilir.

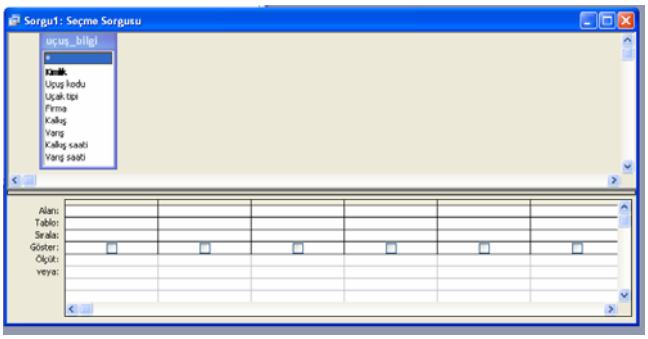
Access programı SQL diline hâkim olmayan ya da bilmeyen programcıların hazırladıkları uygulamalara basit sorgular koyabilmeleri şansını verir. Böylece etkin ve hızlı çözümler sunar. Sihirbaz ile basit sorgular yapılacağı gibi tasarım yöntemi kullanılarak da sorgu oluşturulabilir.

Tasarım için kontrol penceresinde sorgu seçildikten sonra tasarım seçeneği tıklanır. Ekrana sorgu yapılacak tablo seçim penceresi gelir. Yeni bir sorgu sadece tabloları değil aynı zamanda var olan diğer sorguları da kullanabilir.



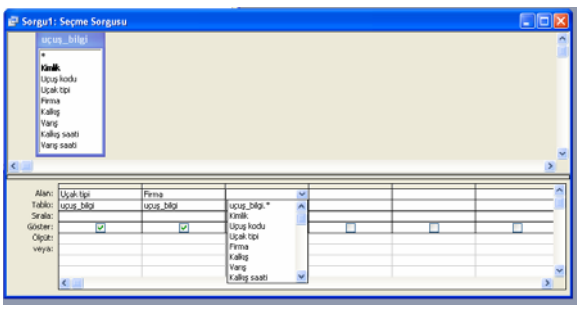
Şekil 1.15: Sorgu penceresi

İlgili tablo (ya da varsa tablolar)seçildiğinde sorgu tasarım ekranının üst kısmında bu tablolar alanlarıyla birlikte yerleşir.



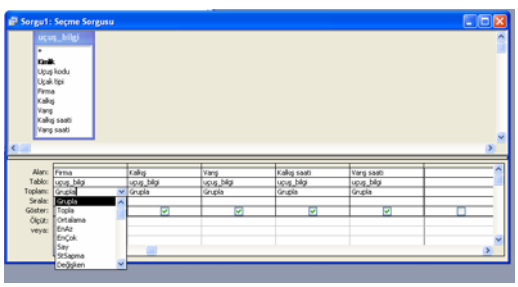
Şekil 1.16: Sorgu penceresine tablonun yerleşimi

Sürükle bırak yöntemiyle ya da sorgu için gerekli alanlar alt kısımdaki alan satırına yerleştirilir. Ya da alan satırlarındaki açılan menüden ilgili alanlar seçilir.



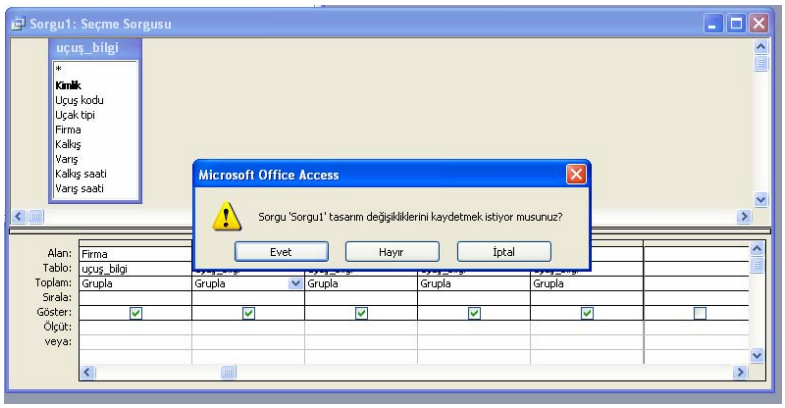
Şekil 1.17: Soru alanlarının yerleşimi

Toplam satırındaki hücrelerdeki açılan menüden o alana ait verilerde yapılacak ve sonucu sorguda gösterilecek işlem seçilir.



Şekil 1.18: Soru alanlarının seçimi

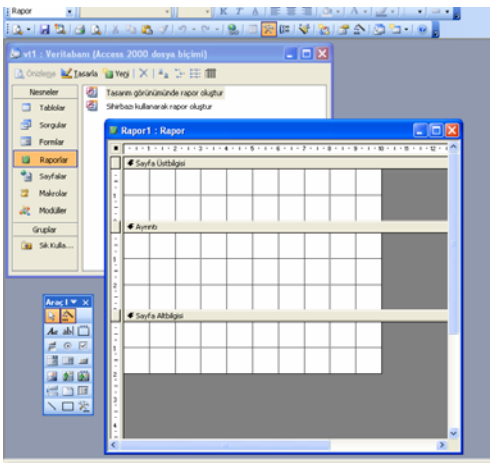
Sırala satırına işlem sonucu verilerin düzeni ve ölçüt satırında ise eğer varsa sorgu için gerekli şartlar girilir. Daha sonra sorgu bir isim verilerek kaydedilir.



Şekil 1.19: Sorgulamanın yapılması

1.2.5. Rapor Oluşturma

Veri tabanı uygulamalarında (tablolarda bulunan) veri yığınlarından sorgu yoluyla toplanan veriler genellikle rapor formatında görüntülenir. İstendiğinde de yazıcı benzeri araçlar vasıtasıyla çıktıları alınır. Bundan dolayı Access programcılarından istenen geliştirdikleri uygulamalarda raporlama işlevinin bulunmasıdır. Veri tabanı kontrol penceresindeki rapor seçeneği programcıya yine tasarım ve sihirbaz olarak iki seçenek hâlinde bu imkânı verir.



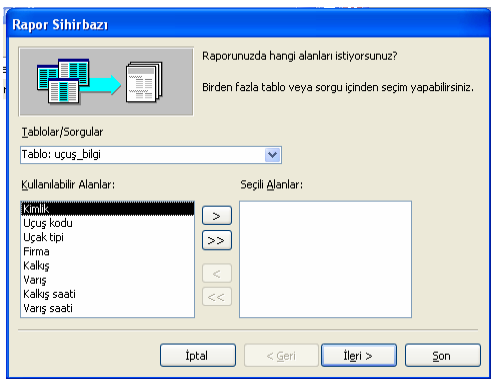
Şekil 1.20: Raporlama penceresi

Tasarım yöntemi form tasarımında izlenen yola oldukça benzer. Özetle ekrana gelen araç kutusundan nesnelerin rapor tasarım penceresine yerleştirilmesi ve ilgili veri bağlantılarının yapılması işlemleri gerçekleştirilir. Yerleştirilen nesnelere ait özelliklerde tıpkı form tasarımında olduğu gibi özellikler penceresi açılarak ayarlanabilir. Rapor tasarım görümü; sayfa üst bilgisi, ayrıntı ve sayfa alt bilgisi olarak üç parçalıdır. Bu parçalarda gösterilmesi istenen veri türleri tespit edilerek planlaması yapılır ve buna göre işlemler yapılır.

ÖRNEK 3:

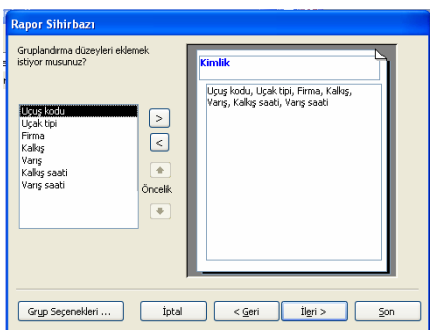
Tekrar havalimanı.mdb uygulamasına geri dönerek uçuş_bilgi tablosuna ait bir rapor oluşturma

Sihirbazı çalıştırıldığında ekrana aşağıda görülen ilk aşama gelir. Bu pencerede raporda görmek istenilen ilgili alanları seçilir. Örnek için firma, kalkış-varış noktaları ile kalkış-varış saatleri rapora eklenir.



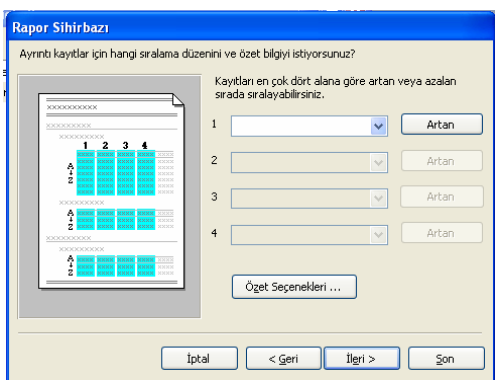
Şekil 1.21: Rapor sihirbazı

Daha sonra aşağıdaki gibi gruplama düzeyi belirlenir.



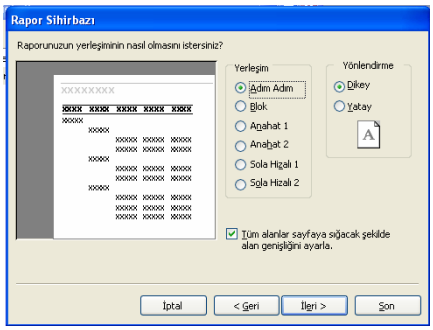
Şekil 1.22: Rapor sihirbazı

Bir sonraki aşamada eğer isteniyorsa sıralama düzeni verilir.



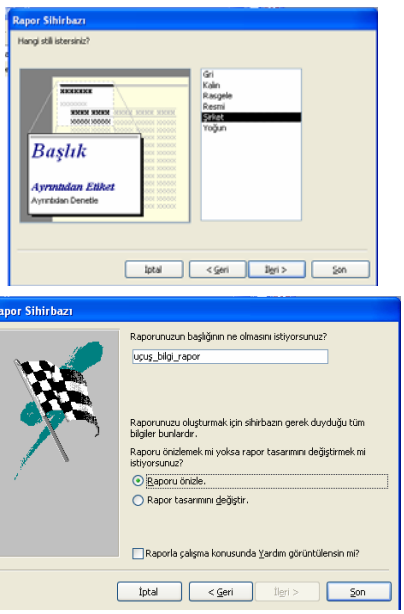
Şekil 1.23: Rapor sihirbazı

Rapordaki verilerin yerleşim düzeni belirlenir.



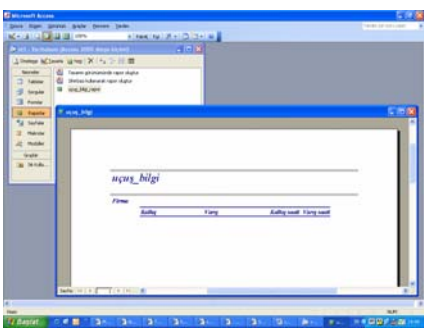
Şekil 1.24: Rapor sihirbazı

Son olarak raporun görünümü ile ilgili seçim yapılır ve rapor kaydedilir.



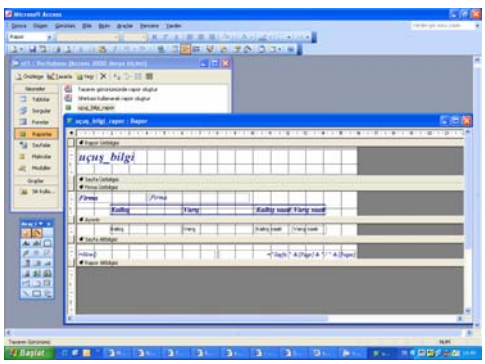
Şekil 1.25: Rapor sihirbazı

Hazırlanan "uçuş_bilgi_rapor" çalıştırıldığında aşağıdaki gibi bir sayfa görüntü elde edilir.



Şekil 1.26: Raporlama

Daha sonradan rapor üzerinde bazı ayarlamalara ihtiyaç duyulursa Access mantığına uygun olarak tasarım görünümünde rapor açılır. Aşağıda görüldüğü üzere rapor alanı üzerinde her türlü değişikliği yapabilecek bir tasarım ekranı karşımıza gelir ve tıpkı form hazırlamada olduğu gibi rapor üzerine yeni kontroller eklenebilir, bunların yerleri ya da biçimleri değiştirilebilir. Bu değişikleri yaptıktan sonra yine kayıt ederek raporlar uygulamanın gerektirdiği hâle getirilir.



Şekil 1.27: Raporlama tasarımı

Bu ders notu sadece Access programını temel ve yüzeysel açıdan incelemeyi ve kullanıcıya accessi tanıştırmayı amaçlamaktadır. Şurası açıktır ki; access veri tabanı programı oldukça geniş seçenekleri kullanıcıya sunan ve detaylarına inildiğinde birçok fonksiyonu içinde barındıran bir programdır. Bu anlamda aslında uzmanlaşmayı gerektirir.

1.2.6. İlişkinin Tanımı

İlişki kısaca iki tablodaki ortak alanlar olarak tanımlanabilir. Saklama alanları arasında (tablolar) var olan ortak alanların belirlenerek bunlar bağlantı kurulması (ilişkilendirilmesi)ile sorgu, form, rapor benzeri işlemlerde oldukça verimli sonuçlar alınır.

Örnek olarak aşağıdaki form dört ayrı tablodan alınan bilgileri içermektedir.

1. Müşteriler tablosu
2. Siparişler tablosu
3. Ürünler tablosu
4. Sipariş Ayrıntıları tablosu

Şekil 1.28: İlişkiler formu

Aynı sipariş hakkındaki bilgiyi yukarıdaki formda olduğu gibi göstermek için bu dört tablonun alanları koordine edilmelidir. Bu koordinasyon, tablolar arasındaki ilişkiler sayesinde yapılır. İlişki, genellikle her iki tablonun aynı adlı alanları olan anahtar alanlardaki veriler eşleştirilerek çalışır. Çoğu durumda, bu eşleştirme alanları, bir tablodan her kayıt için benzersiz bir tanımlayıcı sunan birincil anahtar (Değeri veya değerleri, tablodaki her kaydı benzersiz olarak tanımlayan bir veya birden çok alan (sütun). Birincil anahtar, Null (boş) değerlere izin veremez ve her zaman benzersiz bir dizini olmalıdır. Birincil anahtar, tabloyu diğer tablolardaki yabancı anahtarlarla ilişkilendirmek için kullanılır.) ve diğer tablodaki yabancı anahtardır (Başka bir tablodaki birincil anahtar alanı veya alanlarına başvuran, bir veya birden çok tablo alanı (sütunu). Yabancı anahtar, tabloların ne şekilde ilişkili olduklarını belirtir. Örneğin, çalışan Nu. alanları arasında bir ilişki oluşturarak çalışanlar sorumlu oldukları siparişlerle ilişkilendirilebilir.

Şekil 1.29: Alanlar arasındaki ilişki

1.2.6.1. İlişki Çeşitleri

- **Bir-çok ilişkisi**

Bir-çok ilişkisi en yaygın ilişki türüdür. Bir-çok ilişkisinde, A tablosundaki kayıt, B tablosundaki kayıtta pek çok eşleşen kayda sahiptir, ancak B tablosundaki kayıt, A tablosunda yalnızca bir eşleşen kayda sahiptir.

Şekil 1.30: Bir-çok ilişkisi

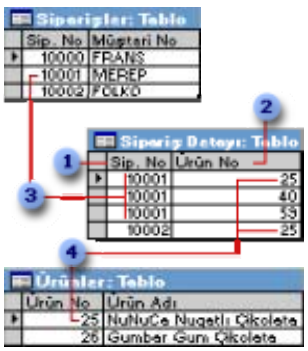
Burada:

- Bir tedarikçi ...
- ... birden çok ürün sağlayabilir ...
- ... ancak her ürünün yalnızca bir tedarikçisi vardır.

Özetlenecek olursa , bir tablodaki her bir veri için ilişkili tablodaki sahaya karşılık gelen birden çok veri varsa buna bir-çok ilişki denir. Daha açık bir örnek olarak; bir babaya karşılık iki çocuk varsa bu bire çok ilişkidir. Çünkü baba her ikisinin de babasıdır.

- **Çok-çok ilişkisi**

Çok-çok ilişkisinde, A tablosundaki bir kayıt, B tablosunda bulunan pek çok kayıtla eşleşebilir ve B tablosundaki bir kayıt A tablosunda bulunan pek çok kayıtla eşleşebilir. Bu tür bir ilişki, birincil anahtarın iki alanı içerdiği, birleşme tablosu olarak adlandırılan üçüncü bir tablonun tanımlanması koşuluyla mümkündür; A ve B tablolarından gelen yabancı anahtarlar örneğin, siparişler tablosu ve ürünler tablosu iki tane bire çok olarak tanımlanan ilişkiyi sipariş detayları tablosunda oluşturarak, çok-çok ilişkilere sahip olur. Bir siparişte birçok ürün bulunabilir ve her ürün birçok siparişte görünebilir.



Şekil 1.31: Çok-çok ilişkisi

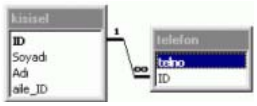
Burada:

- Siparişler tablosundaki birincil anahtar
- Ürünler tablosundaki birincil anahtar
- Bir siparişte birçok ürün bulunabilir ...
- ... ve her ürün birçok siparişte görünebilir.

➢ **Bire-bir ilişki**

Bir tablodaki her bir veri için diğer tablodaki sahaya karşılık gelen bir veri varsa buna Bire-Bir ilişki denir. Örneğin bir kişinin bir saat takması gibi.

Bire-bir ilişkiler bir tablonun kendi sahaları arasındaki ilişkidir. Bire çok ilişkiler ise bir tablonun anahtar sahası ile başka bir tablonun aynı nitelikteki sahası ile ilişkisidir. Bu ilişkilerin en iyi tasarlanabileceği program Ms Access programıdır. Veri tabanı uygulamalarında hiç faydalanılmasa bile sadece ilişkileri düzenleme ekranından faydalanılabilir.



Şekil 1.32: Bire-Bir İlişkisi

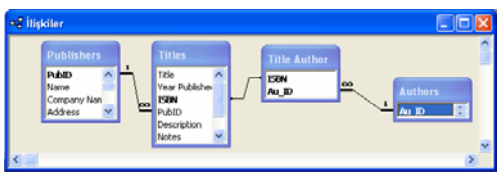
1.2.7. Veri Tabanı Modelleri

Temel olarak 4 çeşit veri tabanından söz edilmesine rağmen en sık kullanılan ilişkisel veri modelidir. Aralarındaki fark verileri nasıl sakladıklarıyla alakalıdır.

Şu anda kullanılan veri tabanlarının çoğu ilişkisel veri modeline daha çok destek verirler. Bu modelde alakalı veriler tablolar içinde saklanır. Ayrıca tablolar arasında değişik türde ilişki kurulur. İlişkiler kurulurken birincil anahtar (Primary key) ve yabancı anahtarlar (Foreign Key) kullanılır. Anahtar alanlar sayesinde indeksleme (indexing) yapma olanağı sunan ilişkisel veri tabanlarında erişim ve işlemler daha hızlı yapılabilir. İlişkisel veri tabanı tasarımı ile aynı verilerin birden fazla kaydedilmesi (ikilenmesi - duplication) önleme şansı vardır. Bu sayede veri tabanını yönetmek daha kolay olduğu gibi verilerin saklama aygıtlarında kapladıkları alanlar da daha küçük olur. Dahası bu sayede veri tabanı ile olan işlemler de daha hızlı gerçekleşir ki veri tabanı işlemleri genelde diğer işlemlere göre masraflı işlemler olarak bilinir.

Tek bir tablo kullanan veri tabanı düz veri tabanı olarak isimlendirilir. Birçok veri tabanı birden fazla tablodan oluşur. Çoklu tabloların kullanıldığı veri tabanı sistemlerinde tabloların ortak bir alanı ortak olmak zorundadır. Bu alanlar aracılığıyla farklı tablolardaki veriler birbiri ile ilişkilendirilir. Bu tip tablolara ilişkili tablolar denir.

Visual Basic ile gelen Biblio.mdb veri tabanı dosyası bilgisayar kitaplarını ve satıcılarının listeleyen örnek bir veri tabanıdır.



Şekil 1.33: Tablolar arasındaki ilişki

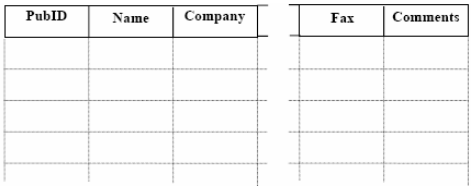
Biblio. mdb dört farklı tablodan oluşmuştur. İlişki yapısını gözden geçirelim.

Authors (yazarlar) tablosu (3 alan, 6246 kayıt)

Au_ID	Author	Year Born

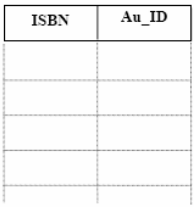
Şekil 1.34: Authors tablosu

Publishers(yayınevleri) tablosu (10 alan, 727 kayıt)



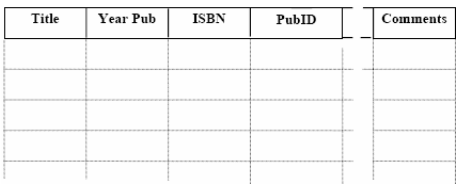
Şekil 1.35: Publishers tablosu

Title author(yazar unvanı) tablosu (2 alan, 16056 kayıt)



Şekil 1.36: Title author tablosu

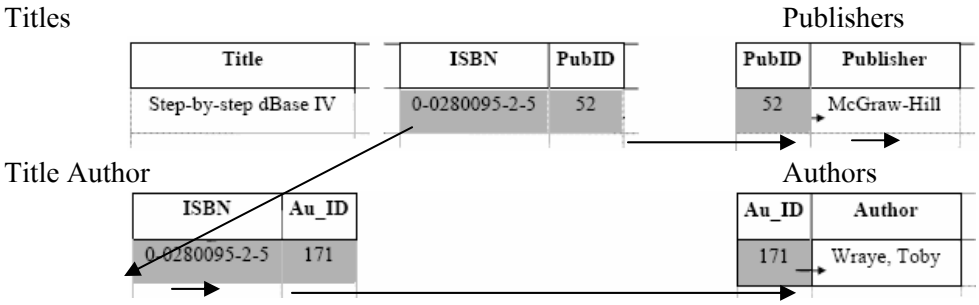
Titles(unvanlar) tablosu (8 alan, 8569 kayıt)



Şekil 1.37: Titles tablosu

Authors tablosu yazar tanımlaması için bir kimlik numarasını, yazar ismini ve doğum tarihini ihtiva eder. Publishers tablosunda kitap yayıncıları konusunda bilgi bulunur. Bir kimlik numarası, yayınevi ismi ve geçerli telefon numarası vardır. Title author tablosu, bir kiatbın ISBN numarası ile yazar kimlik numarasını ilişkilendirir. Titles tablosu kitabın adını, ISBN numarası ve yayınevi numarasını tanımlayan birkaç alanı ihtiva eder.

Burada her bir tablo hem kaynak hem de ilişkisel veriyi barındırmaktadır. Kaynak verisi yazar isimleri gibi verilerdir. İlişkisel veri ise AU_ID ve PubID gibi diğer tablolardaki verilere referans verilerdir. Dört tablodaki ilişkisel veriyi kullanarak veri tabanındaki herhangi bir kitabın tam bir tanıtımı elde edilebilir.



Şekil 1.38: Biblio.mdb'de alanlar arası ilişki

Burada Titles tablosunda yer alan Step-by-step dBase IV kitabının ISBN numarası 0-0280095-2-5 ve PubID numarası 52'dir. 52 sayısını kullanarak Publishers tablosunda yayınevinin McGraw-Hill olduğu görülür. Bu esnada yayınevi ile diğer bilgilere de ulaşılır. Title Author tablosunda ISBN numarası kullanılarak yazarın kimlik numarasının 171 olduğu görülür. Bu bilgi de Authors tablosunda bizi kitabın yazarına götürmektedir.

Veri tabanındaki tüm bilgilerin izlenmesi ve taranması bir "veri tabanı yönetim sistemi" ile tam olarak yapılabilir. Veri tabanını oluşturmak, korumak ve güncellemek için kullanılır. Microsoft Access, Borland Paradox, Borland Dbase, Microsoft FoxPro gibi programlar tek bir bilgisayar üzerindeki veri tabanını yönetir. Microsoft SQLServer, Oracle gibi devasa veri tabanı programları ise ağ üzerinde çalışır.

1.3. ODBC

ODBC (open database connectivity- açık veri tabanı bağlantısı) ilişkisel ve ilişkisel olmayan veri tabanı yönetim sistemlerindeki verilere erişimi sağlayan standart bir API 'dir. Bu API kullanılarak çeşitli bilgisayarlarda farklı veri saklama formatı ve programlama arayüzü kullanan veri tabanı yönetim sistemleri olsa bile veri tabanı uygulamalarının bu verilere erişebilmesi sağlanır.

ODBC, Microsoft'un farklı veri tabanı türlerine(Dbase gibi) bağlantı için geliştirdiği bir standarttır. Farklı veri tabanlarını programlara bağlamak için DSN numarası tanımlanır.

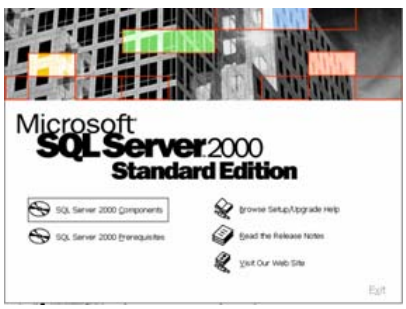
1.4. MS SQL Server 2000

1.4.1. Giriş

Access veri tabanı küçük ölçekli veri tabanı uygulamaları için hızlı, ucuz, kolay bir çözümdür. Ancak veri hacmi10000 civarı bir kayıdın üzerine çıktıktan sonra uygulamalar yavaşlamaya başlar. Microsoft orta ve büyük çaplı uygulamalar için Ms Sql server adıyla yeni bir çözüm sunmuştur. "Ms Sql server" sunucu yazılımı profesyonel anlamda bir hizmet ve etkin bir performans beklenen sistemler için ideal bir yazılımdır. Hâliyle ciddi bir yatırım gerektirir.

MS SQLl server 2000 veri tabanı sunucusunu kurmak bir Office programı kurmak kadar kolaydır. Kurulduktan sonra hemen hemen hiçbir ayar gerektirmeden çalışabilmektedir. Bu açıdan Microsoft ürünlerinin karakteristiğini taşımaktadır.

Kurulum cd'si sürücüye takılarak başlatıldığında sihirbaz aşağıdaki pencereyi getirir. Bu pencerede "SQL Server 2000 Components" seçeneği tıklanarak kuruluma geçilebilir. Bunun altındaki seçenek ise kurulumun yapılabilmesi için önceden bulunması gerekli yazılımları yüklemek için konulmuştur.



Şekil 1.39: SQL server kurulumu

Sonraki aşamada ekrana gelen üç seçenekten ilki olan "install database server" tıklanır.

Şekil 1.40: SQL server kurulumu

Şekil 1.41: SQL server kurulumu

Sonraki gelen pencerede “local computer” seçeneği seçili olmalıdır.

Şekil 1.42: SQL server kurulumu

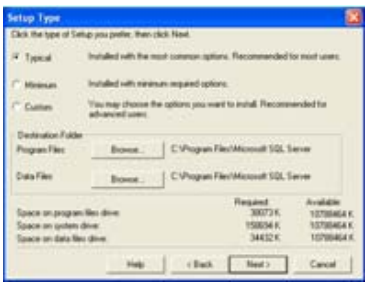
Bundan sonra gelen pencerede “Create a new instance of SQL Server or install client tools” seçeneği seçili durumdadır. Bu şekilde bir sonraki adıma geçilir.

Şekil 1.43: SQL server kurulumu

Sonraki gelen üç pencerede ise firma bilgileri, lisans sözleşmesi ve şifre bilgilerini içeren tipik Windows kurulum adımlarıdır. Standart üç aşama geçildikten sonra ekrana gelen pencere önemlidir. Pencerede ki seçeneklerden “Server and client tools” seçili olmalıdır.

Şekil 1.44: SQL server kurulumu

Standart bir kurulum yapılacağı varsayıldığından typical seçeneği işaretli gelir.



Şekil 1.45: SQL server kurulumu

Daha sonra services accounts penceresine kadar ilerlenir. Bu pencerede seçili olarak gelen “use a domain user account ” yerine “local computer ” seçilir.

Şekil 1.46: SQL server kurulumu

Yetkilendirme ayarları aşağıdaki gibi yapılabilir.

Şekil 1.47: SQL server kurulumu

Sonraki pencerelerde değişiklik yapmadan ilerlenerek kurulum tamamlanır ve aşağıdaki pencere görülerek işlem tamamlanır.

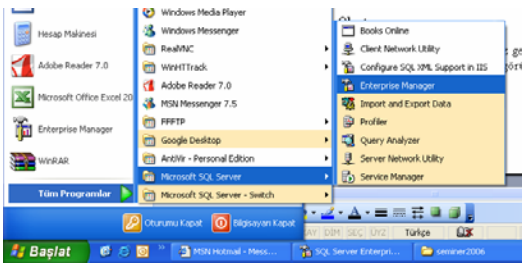
Şekil 1.48: SQL server kurulumu

MS SQL server programı kurulumu tamamlandığında arka planda bir hizmet olarak çalışması gerekir. Bu hizmeti başlatmak için Windows başlat çubuğu sağ köşesinde bulunan ms sql simgesi üzerine çift tıklanarak ekrana gelen pencereden ilgili butonlar tıklanır. Daha kolayı ise bu simge üzerinde iken sağ menüden "start" tıklamaktır. Ya da denetim masası-yönetimsel araçlar – hizmetler bölümünden de benzer işlem yapılır.

Şekil 1.49: SQL server kurulumu

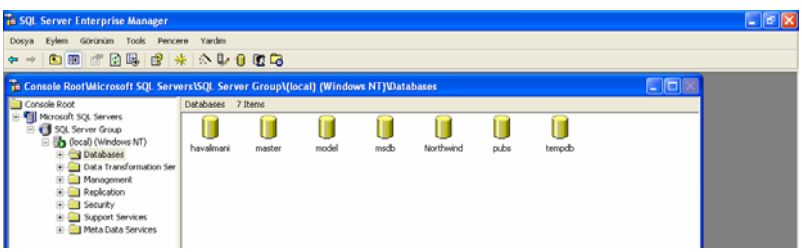
1.4.2. Veri Tabanı, Hesap ve Tablolar Oluşturma

MS SQL server programını makinemize kurduktan sonra yapılması gereken işlem "Enterprise Manager" uygulamasını çalıştırmak olmalıdır. Aşağıdaki şekilde görüldüğü gibi yol izlenerek bu yapılabilir.



Şekil 1.50: 'SQL server'ın çalıştırılması

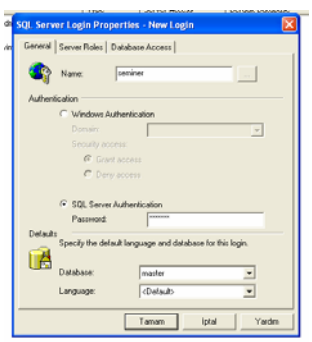
"Enterprise manager" programının ekranı şu şekildedir. Kullanımı oldukça kolay olan bu pencerede yine bir micosoft klasiği olan "sol kısım agaç, sağ kısım içerik" yapısı kullanılmıştır.



Şekil 1.51: SQL Server ortamı

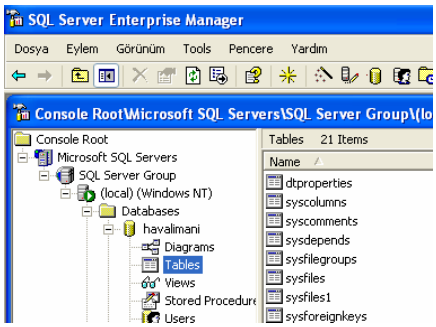
- Yeni bir veri tabanı oluşturma: Sol kısımdan "SQL server group" ifadesinin başındaki kutucuk tıklanarak 'local'a ulaşılır. Local'in altına girilerek Databases kısmına erişilir. Databases tıklanır ve sağ tarafta bol bir alanda sağ menüden "new database" seçilir. Ekrana gelen pencerede oluşturulacak yeni veri tabanına isim veririr. (Örnek olarak "havalimanı" veriniz.). Tamam tıklandığında havalimanı veri tabanının oluştuğunu göreceğiz.

- Hesap oluşturma: Yeni bir veri tabanı oluşturduktan sonra erişim denetimi yapmak için sadece bu veri tabanı için bir kullanıcı oluşturulur. Bu oldukça kolay bir işlemdir. Öncelikle sol kısımda "security" klasörünün altına girerek "logins" seçilir. Sağ tarafta bol bir alanda sağ fare menüsünden "New login" seçeneği tıklandığında karşımıza aşağıdaki ekran gelir. Burada oluşturulacak yeni kullanıcı için isim girilir. (örnek: seminer). "Sql server authentication seçerek şifre verelim. (örnek: seminer). Alt kısımda database bölümüne daha önceden oluşturulan havalimanı, language olarakta Türkçe seçilir. Son olarak database Access sekmesini açılarak bu veri tabanı listesinden 'havalimanının' başındaki kutucuğunu ve alt kısımdan bu veri tabanın sahibinin bu kullanıcı olduğunu belirten "db_owner" işaretlenir. Tamam tıklandığınında ekrana gelen onay penceresine az önce girdiğimiz şifreyi (seminer) girerek havalimanı veri tabanı için yeni bir hesap açılmış olunur.



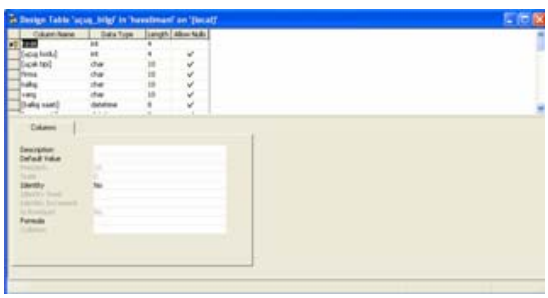
Şekil 1.52: SQL Server'a bağlanma

- Yeni bir tablo oluşturmak: oluşturulan veri tabanında tablo oluşturmak için sol kısımdaki database altına girerek havalimanı veri tabanı açılır. Burada tables seçerek sağ tarafta boş bir alanda sağ fare menüsünden "New Table" tıklanır.



Şekil 1.53: SQL Server hesap açma

Ekrana aşağıdaki gibi bir pencere gelir. Dikkatli incelendiğinde bu pencerenin Access tablo tasarım penceresiyle oldukça benzerlik gösterdiği fark edilebilir. Bu pencerenin üst kısmına tabloda yer alacak veri alanları ve tipleri girilir. Eğer detaylı ayrıntılar isteniyorsa alt kısımda bulunan özelliklerden faydalanılabilir. Tıpkı Access 'te olduğu gibi bu özellikler bölümü her veri tipi için farklılık arz etmektedir. Girilen alanlardan hangisine "birincil anahtar" vermek isteniyorsa o alanın bulunduğu satırda sağ fare menüsünden "set primary key" tıklamak yeterlidir. Eğer araya yeni alanlar girmek ya da silmek isteniyorsa aynı menüden "insert column"/ "delete column" kullanılır.



Şekil 1.54: SQL Server hesap açma

Buraya kadar anlatılanlar MS SQL server sunucusunun en temel özelliklerini içermektedir. MS SQL server 'da veri tabanı, tablolar ve yetkilendirme yaptıktan sonra visual basic kodları vasıtasıyla erişim sağlanarak uygulamalar geliştirilebilir.

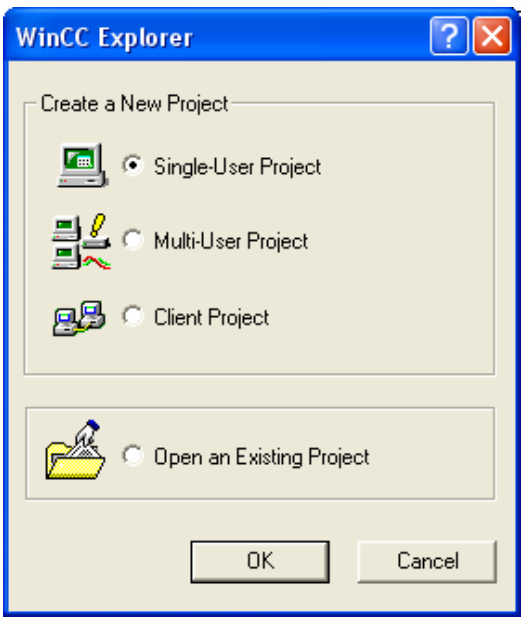
SQL sunucusunun detay ayarları ve seçenekleri ile oldukça esnek hareket kabiliyetine sahip olunulabilir. Örneğin Access veri tabanından veri çekme (import), diğer veri tabanı sunucuları ile bağlantı kurma (link) işlemler kolaylıkla yapılabilir.

Bu bağlamda Ms Sql server sunucusu RDBM (ilişkisel veri tabanı yönetimi) alanında rakipleriyle boy ölçüşecek kadar güçlü çözümdür.

1.5. WinCC Ortamını Tanıma

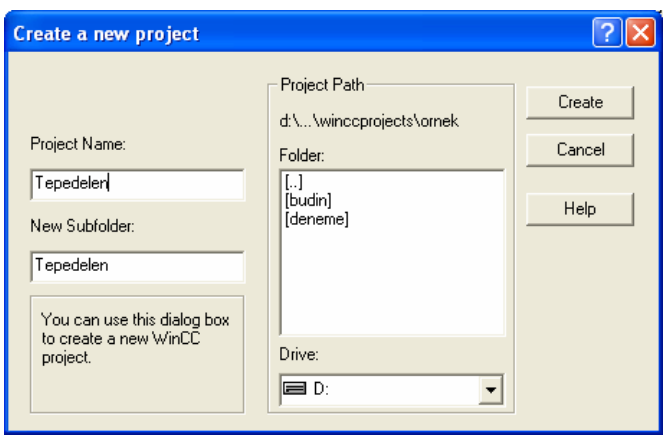
- **Proje oluşturma**

"File-new" menüsünden yeni bir proje seçilir. WinCC projesi sadece tek kullanıcılı olacağı için "Single-User Project" seçilir.



Şekil 1..55:Yeni proje oluşturma

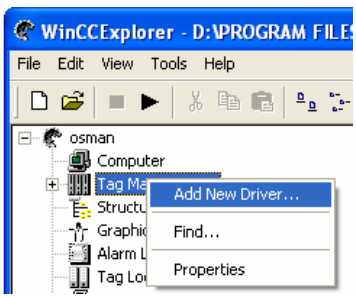
"Project name" alanına bir isim yazılarak "create" düğmesine basılır. Proje ismi farklı isimde bir klasör altında oluşturulmak istenirse "new subfolder" alanına bu isim girilir.



Şekil 1.56:Yeni proje oluşturma

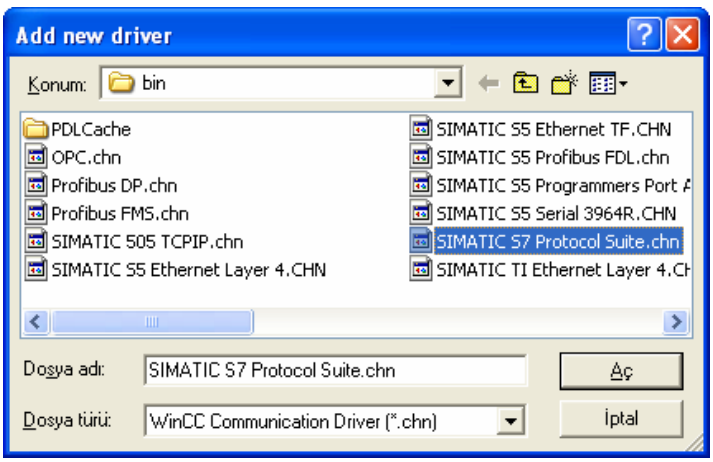
➢ **Projeye Plc'nin eklenmesi**

"Tag management" seçilir ve farenin sağ tuşuna basılır. Açılan menüden "add new driver" seçilir.



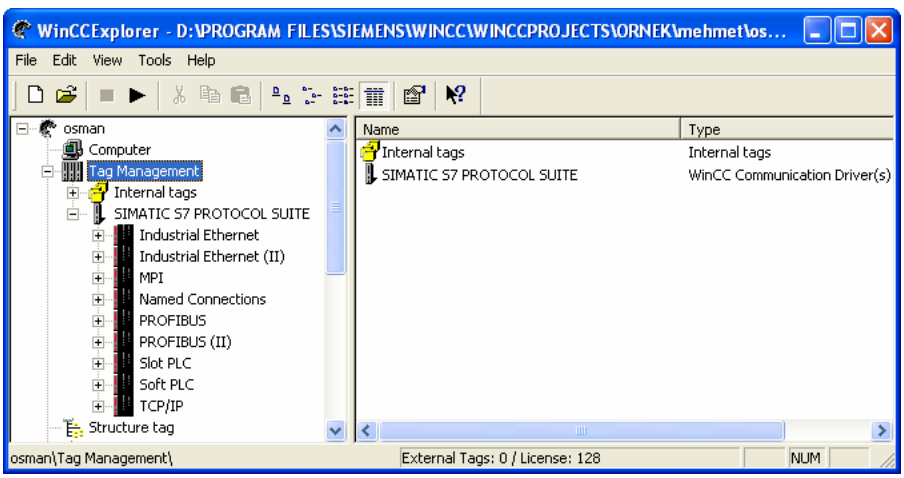
Şekil 1.57:Yeni sürücü oluşturma

S7 300 PLC ile çalışılacağı için "SIMATIC S7 protocol suite" seçilir.



Şekil 1.58:Yeni sürücü oluşturma

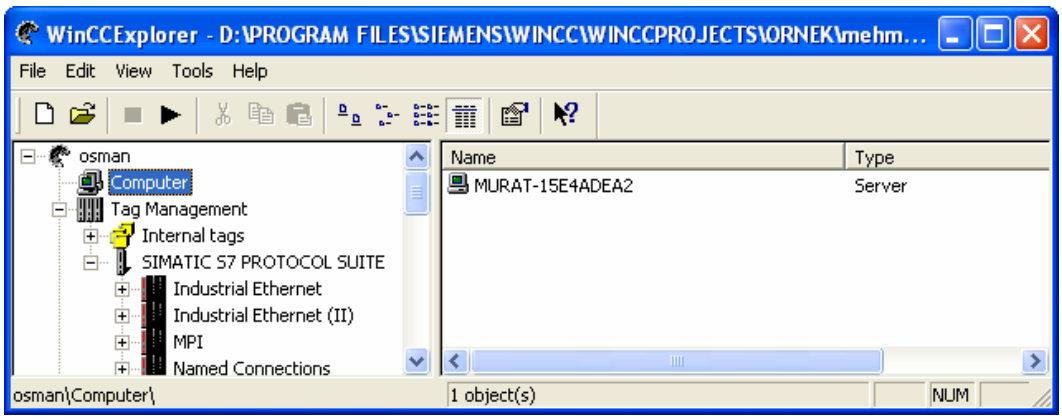
"Tag management" alanı altında "SIMATIC S7 protocol suite" başlığı oluşturulur. Bunun altında ise PLC CPU'sunun haberleşme şekilleri görünür.



Şekil 1.59:Haberleşme protokolü oluşturma

Burada “MPI” ile haberleşilecekse MPI üzerinde farenin sağ tuşu tıklanarak “new driver connection” seçilir. Projeyi çalıştıracak programlama aygıtı seçilmiş olur.

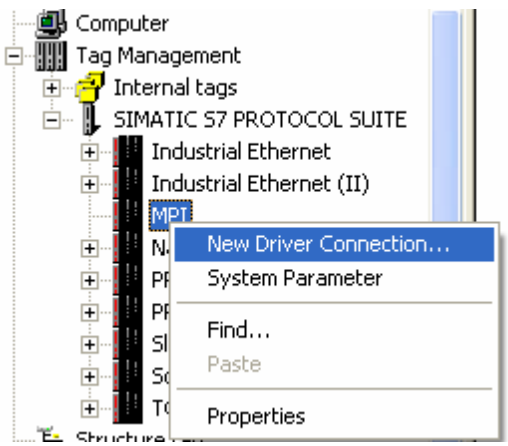
Farklı bir bilgisayarda hazırlanmış WinCC projesinin diğer bir bilgisayarda çalışması için “computer” alanında, isim değişikliği yapmak gerekir. Bunun için bu alan işaretlenir.



Şekil 1.60: Server değişimi

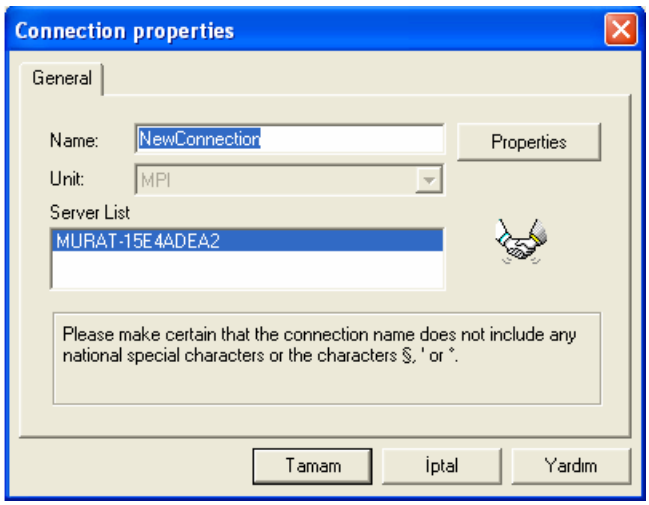
Sağ sütunda projede kayıtlı bilgisayar ismi görünür. Üzerine çift tıklanarak ya da farenin sağ tuşu ile ”properties” özelliği seçilerek açılan pencerede, “general-computer name” alanına, projenin üzerinde çalıştığı bilgisayarın ismi girilir.

MPI üzerinde iken farenin sağ tuşuna tıklanarak “new driver connection” seçilir.



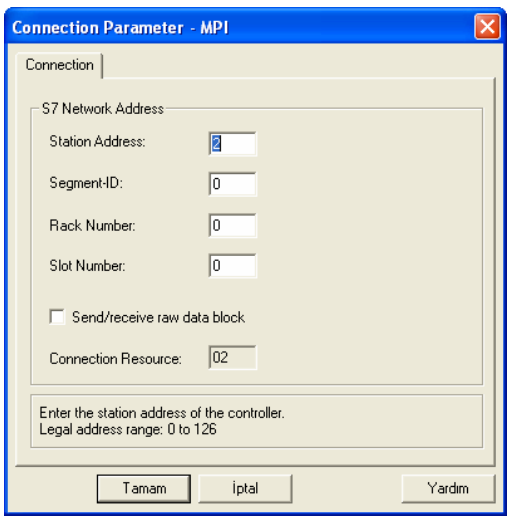
Şekil 1.61: Yeni bağlantının oluşturulması

Bağlantı adına bir isim verilerek “properties” düğmesine tıklanır.



Şekil 1.62: Yeni bağlantının oluşturulması

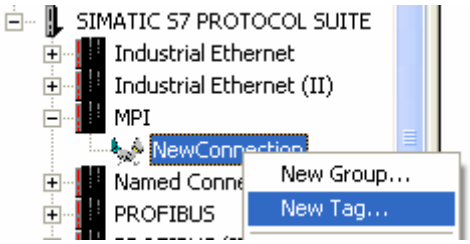
Projede kullanılan CPU’nun ray numarası, istasyon adresi girlir.



Şekil 1.63: Yeni bağlantının oluşturulması

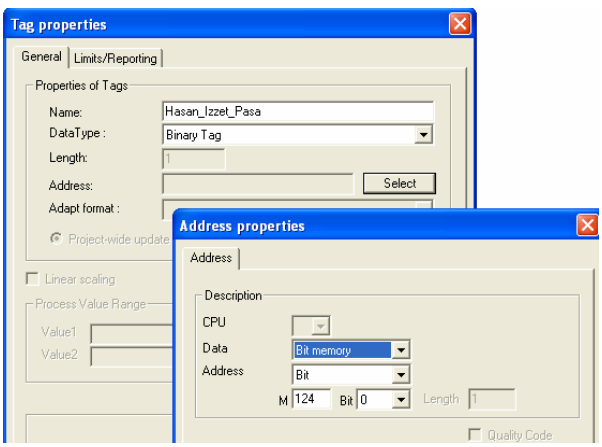
- **Tag Eklenmesi**

PLC’nin giriş-çıkış uçlarını ya da dâhilî bitlerini temsil etmek için SCADA programlarında tag(etiket) kullanılır. Yukarıdaki işlem sonucunda “MPI” alanının altında “new connection” ya da verilmiş olunan bir isimde bağlantı oluşur. Yeni bir etiket oluşturmak için sağ tuşa tıklanır.



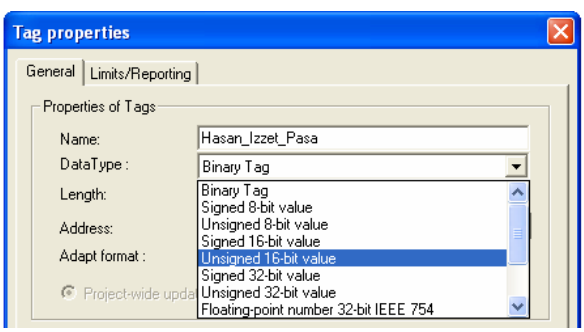
Şekil 1.64: Yeni bir etiketin oluşturulması

Tag properties penceresinde "data type" alanına ikilk tipte bir etiket girilmesi isteniyor. "Select" düğmesine tıklanarak "data" alanına "bit memory" ve "adres" alanına 124.0 (ya da yapılan ayarlamaya göre 0.0)girilir.



Şekil 1.65: Yeni bir etiketin oluşturulması

"Data type" alanına bakıldığında, atanabilecek diğer değerler görülmektedir. Örneğin işaretsiz sayısal bir değer için "Unsigned 16-bit value" seçilir.



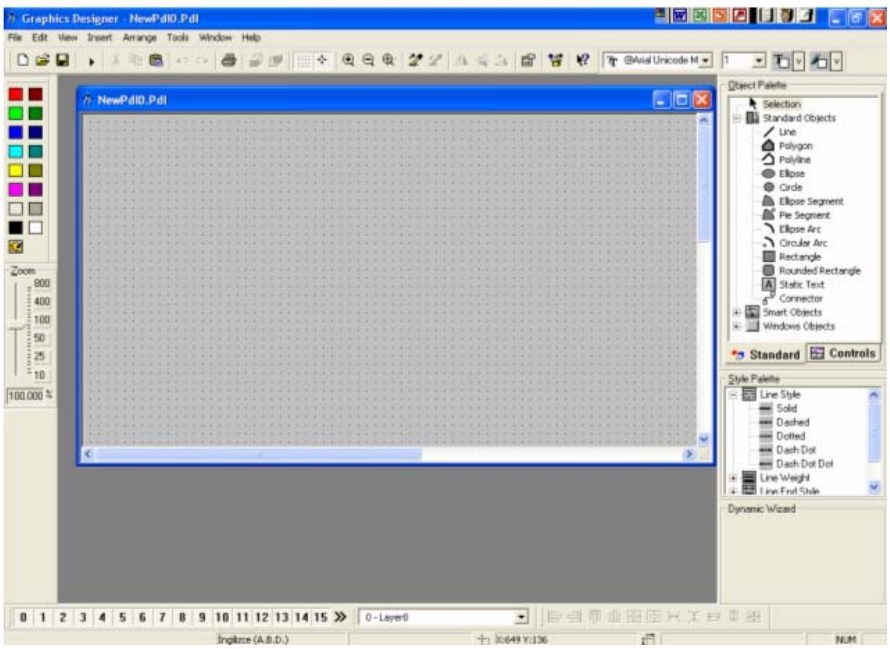
Şekil 1.66: Yeni bir etiketin oluşturulması

- **Grafik Eklenmesi**

Yapılacak işe göre grafik ekran tasarlanır. Hareketli ya da durağan olarak tasarlanabilen elemanlar vasıtasıyla PLClerle veri alış verişi sağlanabilir. Birden fazla pencere tasarlanabilir.

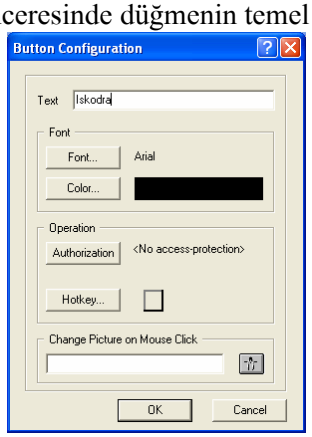
"Graphic designer" üzerinde sağ tuş tıklanarak "new picture" ile yeni pencere oluşturulur. Bu pencereye varsayılan olarak "newPdl0.pdl" ismi verilir. Farenin sağ tuşuna tıklanarak "rename picture" ile yeni isim verilebilir. Pencereyi açmak için çift tıklanır ya da sağ tuş tıklanarak "open picture" işaretlenir.

Açılan pencerenin sağ tarafında "object palette" yer alır. "Standart objects" ile dikdörtgen gibi temel geometrik şekiller çizilebilir. "Smart objects"de resim, video pencere getirilebilir. "Windows objects" ile düğme, onay kutusu gibi temel windows bileşenleri pencereye eklenebilir.



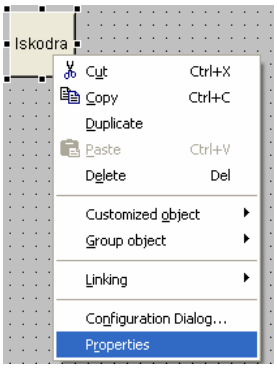
Şekil 1.67: Grafik ekran

Şimdi projeye çift tıklayarak bir düğme ekleyelim. Butonun pencere yerleşme anında açılan "button configuration" penceresinde düğmenin temel özellikleri ayarlanır.



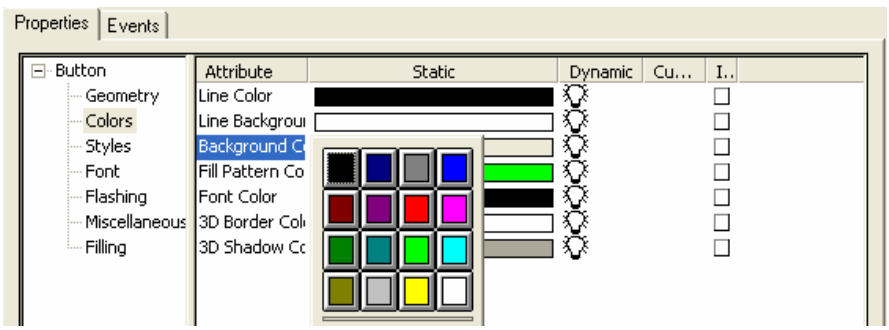
Şekil 1.68: Bir butonun oluşturulması

Düğmenin özellikleri açılır.



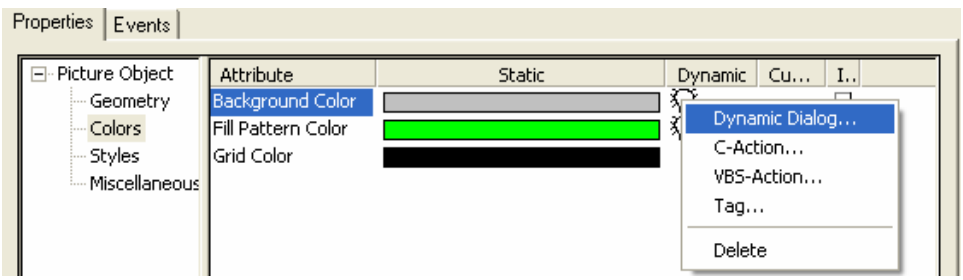
Şekil 1.69: Düğmenin özellikleri

Açılan "object properties" penceresinin "properties" alanında colors özelliğine ait "background color" seçilir. "Static" sütununda bulunan renk, çift tıklanarak renk değiştirilir.



Şekil 1.70: Düğmenin özellikleri

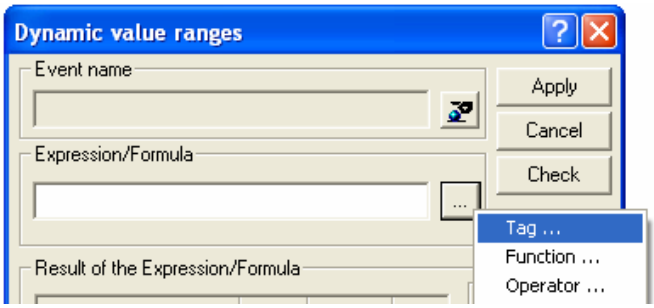
Düğmenin çalışma anındaki rengini yani dinamik hâldeki özelliklerini değiştirmek için bu düğmenin daha önce hazırlanmış etikete atanması gerekmektedir.



Şekil 1.71: Düğmenin özellikleri

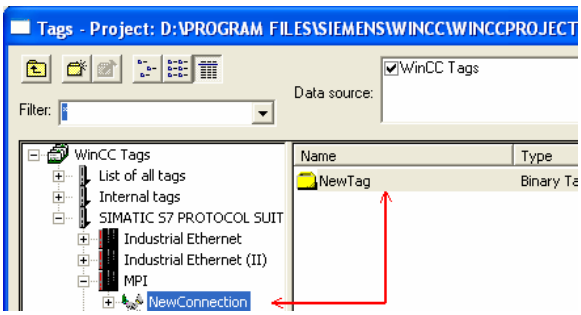
Dinamik özelliği değiştirilmek istenen niteliğin yanındaki ampül şekline sağ tıklanarak "dynamic dialog" seçilir.

Açılan pencerede "expression/formula"nın yanındaki düğmeden "tag" seçilir.



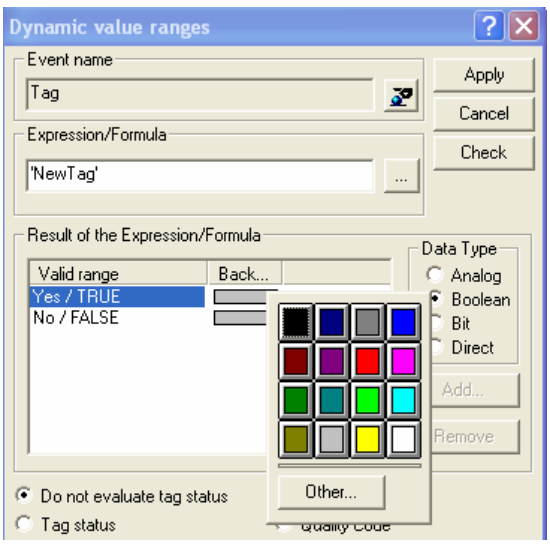
Şekil 1.72: Düğmeye etiket atanması

Daha önce oluşturulan etiket atanır.



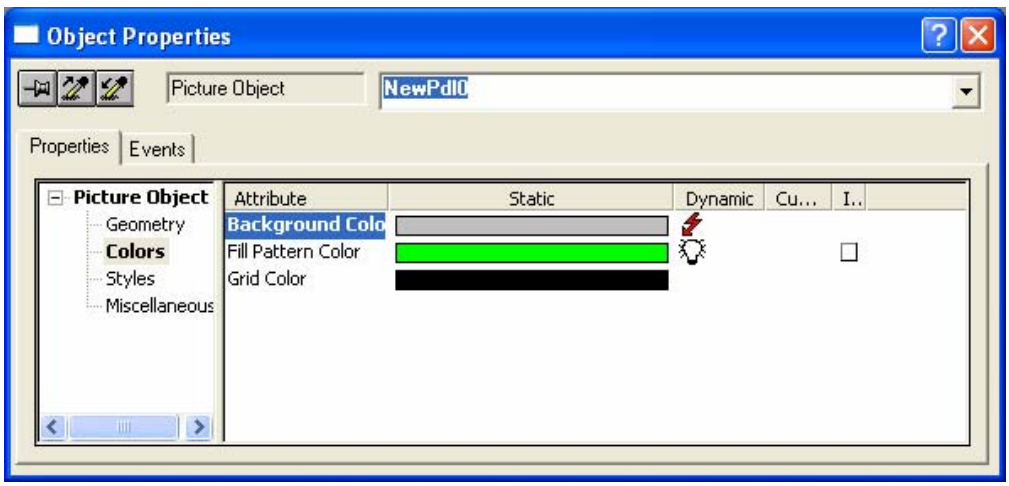
Şekil 1.73: Düğmeye etiket atanması

Atanan etiket Boolean tipinde tanımlanmıştı. Bunun için "Result of the Expression/Formula" alanında "data type" olarak Boolean seçilir. "Valid range" sütununda, düğmenin basıldığında ve bırakıldığında alacağı renk değerleri verilir.



Şekil 1.73: Düğmeye etiket atanması

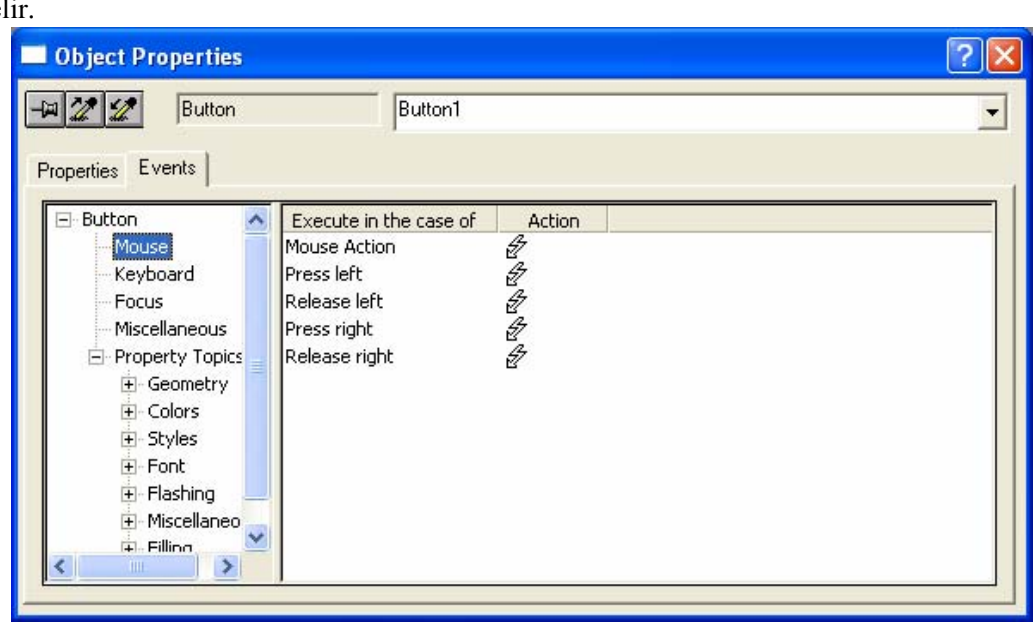
"Apply" düğmesine tıklanır. "Object properties" penceresinde özelliği değiştirilen rengin Dynamic sütününda şimşek şekli belirir.



Şekil 1.74: Düğmeye etiket atanması

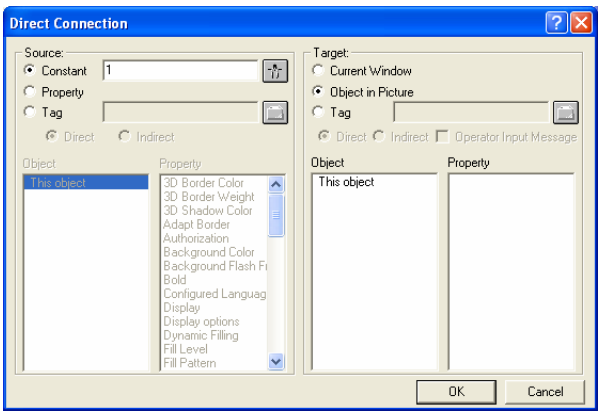
Son olarak düğmeye bir olay ataması yapılarak basıldığında ve serbest bırakıldığında ne yapması gerektiği belletilir. Burada düğme, bir anahtar görevi yapacaktır. Basıldığında "1", serbest bırakıldığında "0" değerini alacak ve bu değerler PLC'deki programı etkileyecektir.

"Object properties" penceresinde "events" sekmesi tıklanır. Düğmeye fare ile hareket verileceği için "mouse" seçilir. Sağ tarafta bu esnada farenin sağ-sol tuşlarına ait olaylar gelir.



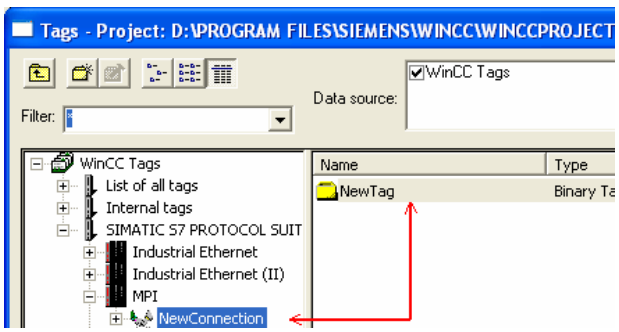
Şekil 1.75: Olay atanması

“Press left”’in karşısındaki şimşek timsali çift tıklanır. Düğmeye basıldığında değerinin “1” olması için “source” alanında “constant” onay kutusuna “1” yazılır. “Target” alanında “tag” işaretlenir.



Şekil 1.76: Olay atanması

Sağ tarafta beliren çanta sembolü tıklanarak daha önce oluşturulan etiket atanır.

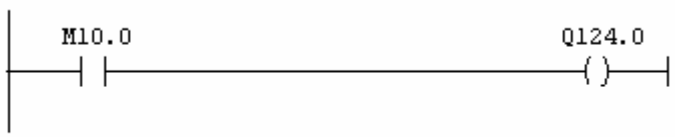


Şekil 1.77: Olay atanması

Aynı işlem “Release left” için tekrarlanır. Tek fark “constant” değerine 0 atanır.

- **PLC Programı**

Burada PLC’ye ait 124.0 nu.lı çıkış ucu SCADA arayüzü üzerinden sürülmek isteniyor.



Şekil 1.78: PLC Programı

Aynı çıkış hem PLC'den hem de oluşturulan SCADA arayüzü üzerinden sürülmek istenirse "VEYA" kapısı kullanılır.

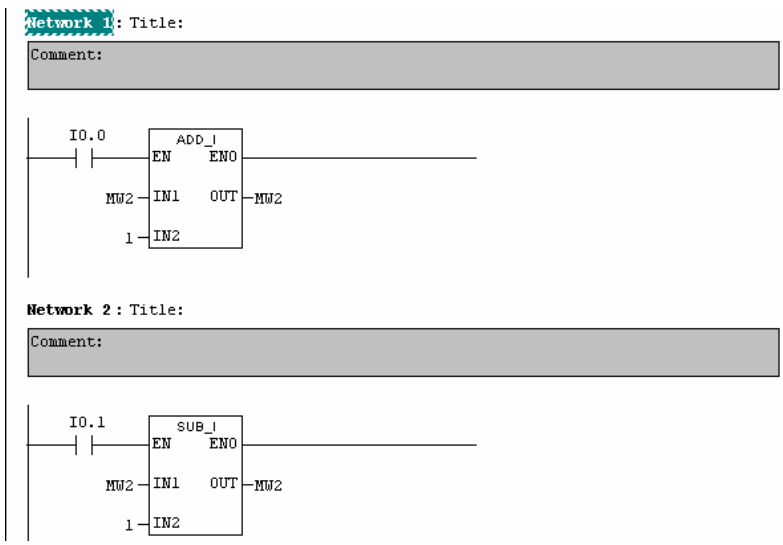
Şekil 1.79: PLC Programı

Bu aşamada SCADA arayüzü kaydedilir ve çalıştırılır.

1.6. WinCC'de Verileri Görme

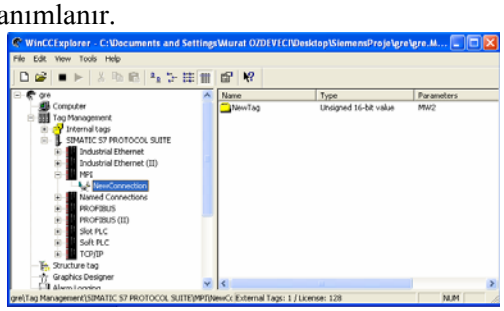
WinCC'de "Tag Logging"i görme

- PLC program yazılımı örneği aşağıdaki gibidir.



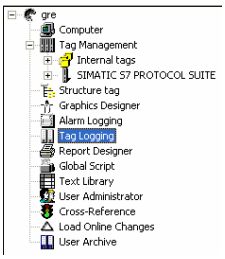
Şekil 1.80: PLC programı

- PLC ile WinCC arasındaki bağlantıyı yaparak depo değerini tutan değeri ifade edecek tag tanımlanır.



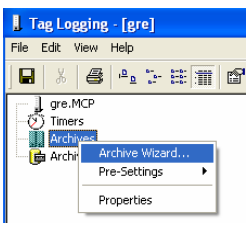
Şekil 1.81: Etiket "tag" oluşturma

➢ Tag Logging çalıştırılır.



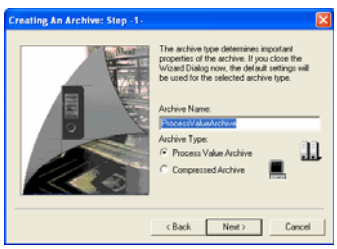
Şekil 1.82: Tag logging

➢ Erana gelen pencerede "archives" üzerinde sağ tıklayarak "archine wizard" seçilir.



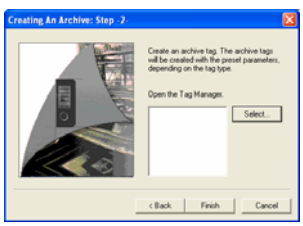
Şekil 1.83: Arşiv sihirbazı

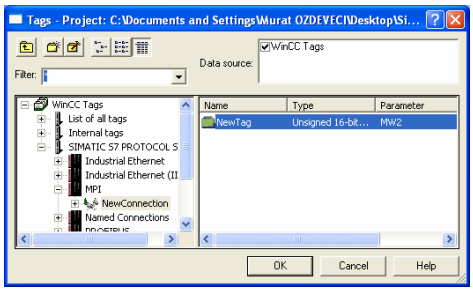
Arşiv ismi yazılır. Tip olarak "process value archive" yazılır.



Şekil 1.84: Arşiv sihirbazı

Ekran gelen pencerede "select" düğmesi tıklanarak daha önce hazırlanan "tag" ismi işaretlenir.

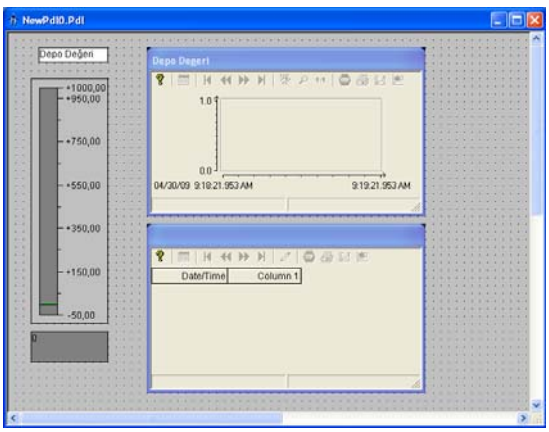




Şekil 1.85: Etiket belirleme

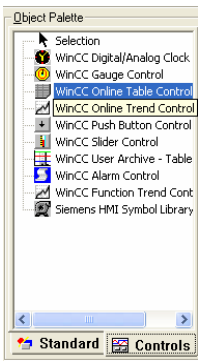
Onay verdikten sonra ayralar kaydedilir.

- "Graphic designer"de yeni bir sayfa oluşturarak sayfa tasarımı yapılır.



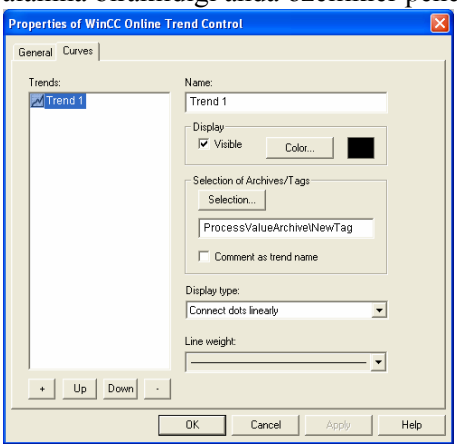
Şekil 1.86: Sayfa tasarımı

"Trend ve table control" bileşenleri "controls" sesinin altındadır.



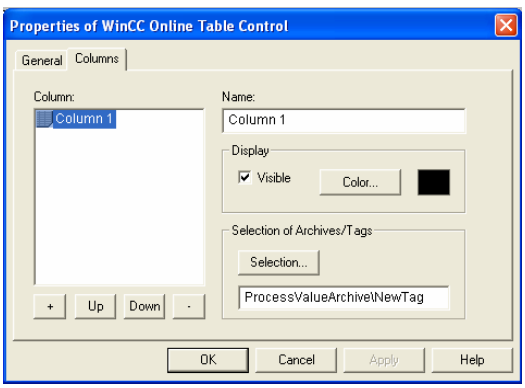
Şekil 1.87: Sayfa tasarımı

Trend Control resim alanına bırakıldığı anda özellikler penceresi ortaya çıkar.



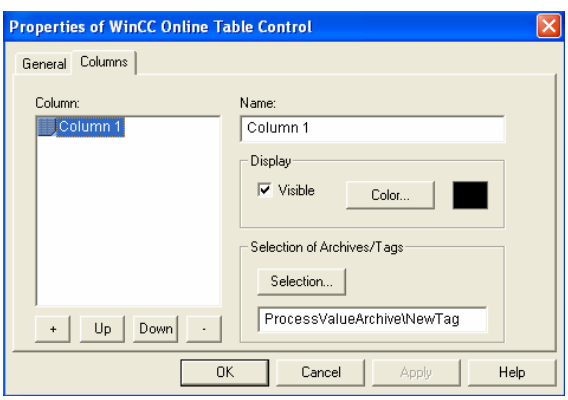
Şekil 1.88: Trend kontrol nesnesi özellikleri

Curves sekmesi altında "selection" düğmesi tıklanarak daha önce oluşturulan "tag" bulunur.



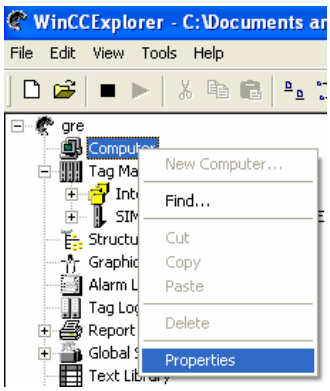
Şekil 1.89: Trend kontrol nesnesine etiket bağlama

"Table Control" resim alanına bırakıldığı anda özellikler penceresi görünür. "Columns" sekmesi altında "selection" düğmesi tıklanarak daha önce oluşturulan "tag" bulunur.



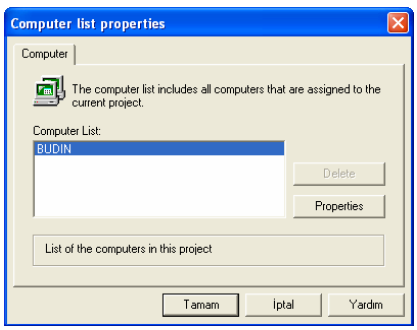
Şekil 1.90: Trend kontrol nesnesine etiket bağlama

➢ WinCC'nin ana penceresine geçilir. Computer'in özelliklerine girilir.



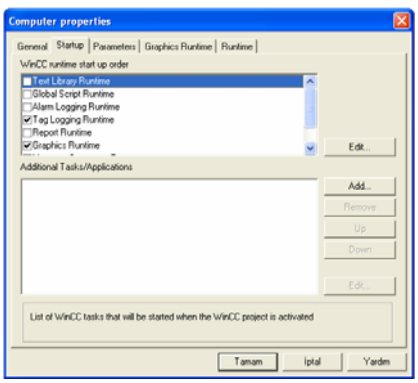
Şekil 1.91: Bilgisayar listesi

Properties düğmesine tıklanır.



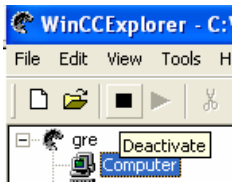
Şekil 1.92: Bilgisayar seçimi

"Startup" sekmesine "Tag logging runtime" işaretlenir. Onay verilir.



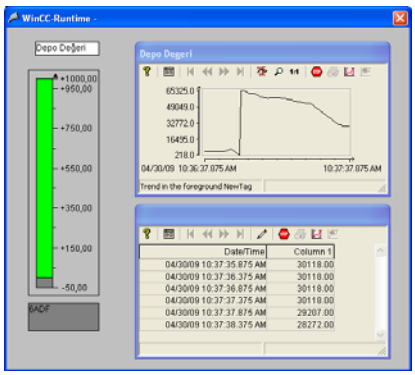
Şekil 1.93: Tag logging runtime

Projenin çalışması durdurulur ve tekrar başlatılır.



Şekil 1.94: Tag logging runtime

- Grafik ekrana geçerek program çalıştırılır.



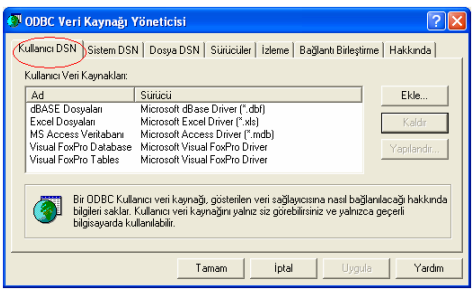
Şekil 1.95: Tag logging runtime

ÖRNEK 4: WinCC'den Access'e veri kaydetme

Access'te birincil anahtar görevi verilen ID ve Deger isimli iki alan oluşturulabilir. Tablonun ismi Tablo1, veri tabanının ismi Data olabilir.

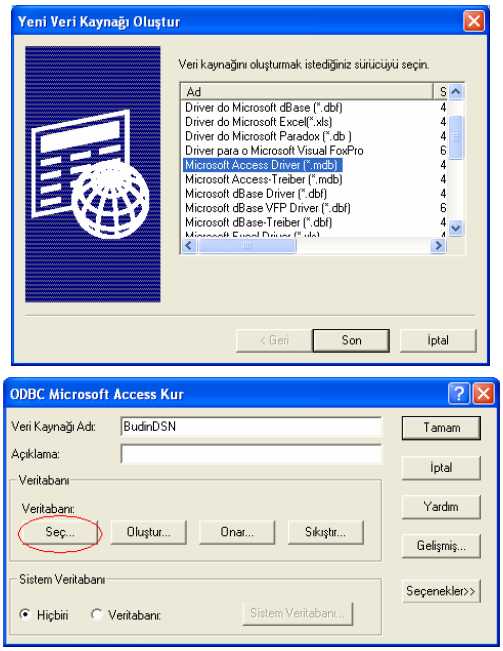
Şekil 1.96: Acces'te tablo oluşturma

Denetim masası/yönetimsel araçlarından veri kaynakları (ODBC) açılır. Oluşturulan bu veri tabanı için bir kısayol (DSN: data source name) oluşturulur.



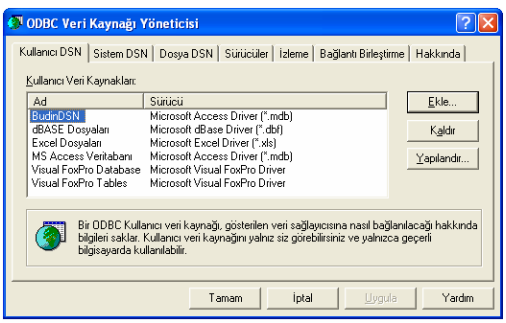
Şekil 1.97: ODBC paylaşımı açma

Ekle düğmesine tıklanır. Access sürücüsü seçilir.



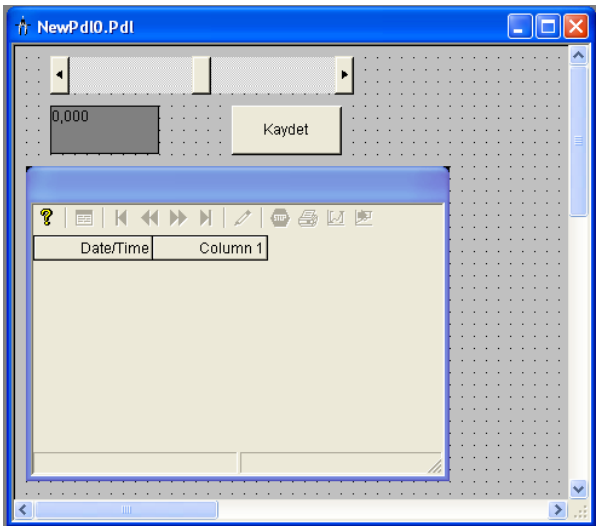
Şekil 1.98: ODBC paylaşımı açma

Veri kaynağı adına bir isim verilir ve “seç” düğmesine basılarak bağlantı kurulacak veri tabanı(biraz önce oluşturulan Access veri tabanı) bulunur. Tamam denilir.



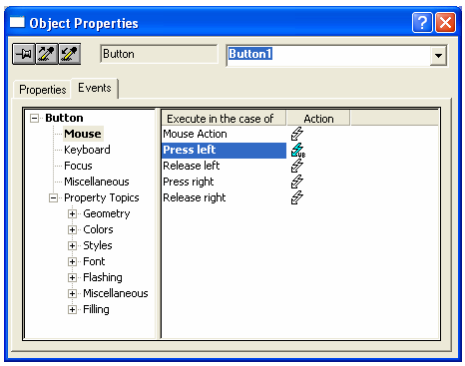
Şekil 1.99: ODBC paylaşımı açma

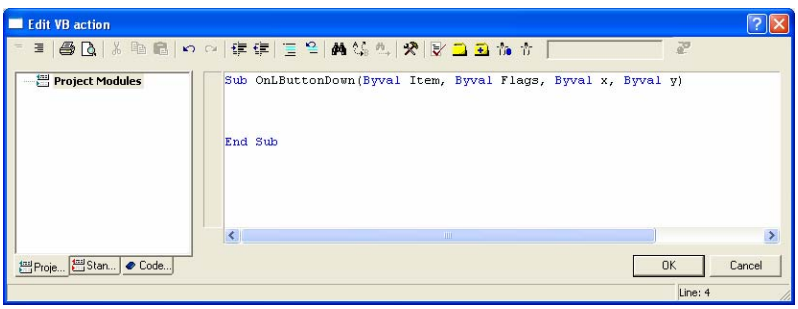
Yukarıda oluşturulan WinCC projesi açılarak grafik ekranda, daha önceki elemanların yanına bir düğme eklenir.



Şekil 1.100: WinCC ekran tasarımı

Düğmeye sağ tuşla tıklanarak özellikler penceresine girilir ve "events" sekmesinden Mouse için "press left" olayı çift tıklanır.





Şekil 1.101: WinCC kod ekranı

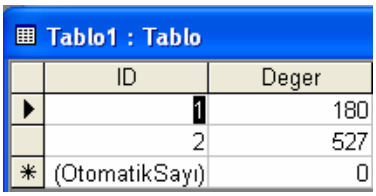
Farenin

```
OnLButtonDown
        End Sub
```

olayı arasına aşağıdaki kodlar eklenir.

```
Dim objConnection

Dim strConnectionString

Dim Zenta

Dim strSQL

Dim objCommand

strConnectionString =
"Provider=MSDASQL;DSN=BudinDSN;UID=;PWD=;"

Zenta = HMIRuntime.Tags("NewTag").Read

strSQL = "INSERT INTO Tablo1 (Deger) VALUES (" & Zenta &
");"

Set objConnection = CreateObject("ADODB.Connection")

objConnection.ConnectionString = strConnectionString

objConnection.Open

Set objCommand = CreateObject("ADODB.Command")

With objCommand

.ActiveConnection = objConnection

.CommandText = strSQL

End With

objCommand.Execute

Set objCommand = Nothing

objConnection.Close

Set objConnection = Nothing
```

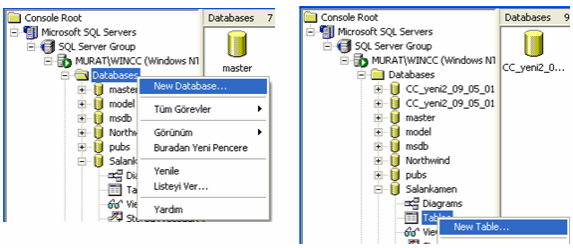
Burada NewTag, kaydırma çubuğundan elde edilen değeri tutmaktadır. Kaydırma çubuğunu oynattıktan sonra “kaydet” düğmesine basılır. Access veri tabanının görüntüsü:



Şekil 1.102: Tablo içindeki veriler

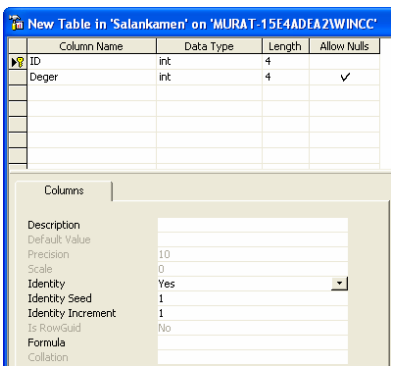
ÖRNEK 5: WinCC-SQL Server bağlantısı

SQL Server'da vir tablo oluşturma



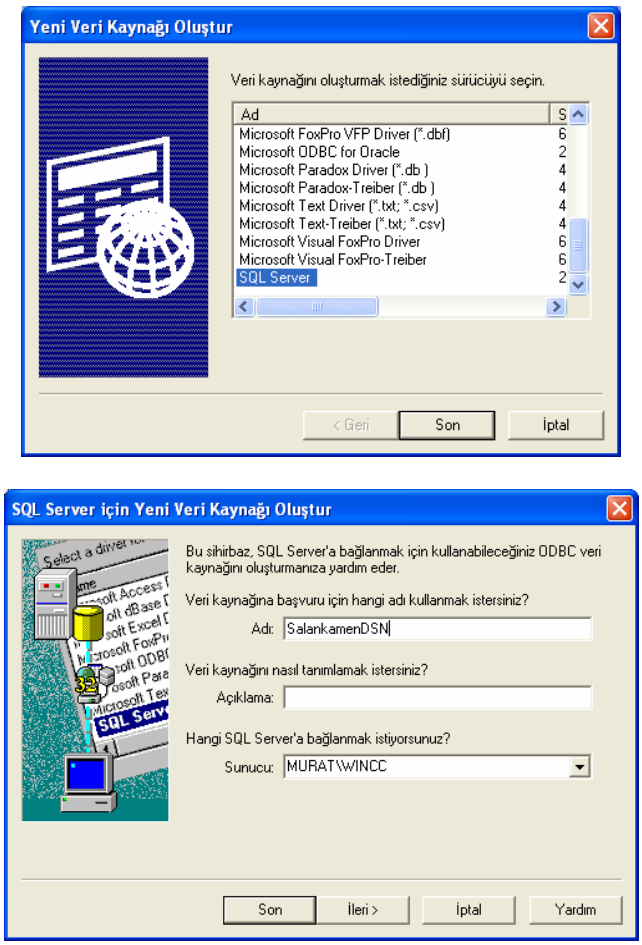
Şekil 1.103: SQL Server'de tablo oluşturma

Diğer Access tablolarında olduğu gibi iki alan girilir. ID2nin Identity özelliği "yes" olarak ve tabloyu "Tablo1" olarak kaydedilir.



Şekil 1.104: SQL Server'de tablo oluşturma

Yeni bir DSN oluşturulabilir.



Şekil 1.104: Veri kaynağı oluşturma

Bir önceki örnekteki kodta DSN ismini değiştirerek yeniden yazılır.

UYGULAMA FAALİYETİ

Aşağıda verilen işlem basamakları ve öneriler doğrultusunda verilen veritabanı uygulamasını yapınız

İşlem Basamakları	Öneriler
Daha önce öğrendiğiniz veri tabanı programını hatırlayarak aşağıda size verilen kriterlere göre bir veri tabanı oluşturunuz.	Access veri tabanı programı size örnek olabilir.
Oluşturulacak veri tabanı, bir derneğe kayıtlı üyelerin kişisel bilgilerini tutan bir veri tabanı olacaktır. Kişisel bilgileri içeren alanları isteğinize uygun olarak düzenleyebilirsiniz.	Kişisel bilgileri içeren alanlar şunlar olabilir. T.C. Kimlik Nu. Ad_Soyad Dog_Yer Dog_Tarih Adres Telefon Medeni_Hal Cocuk_Sayi Egitim_Dur Meslek
Size verilen bu alan adlarını dikkate alarak aşağıda verilen tablodaki boşluklara alan adlarını, alanın veri türlerini ve alanın uzunluğunu yazınız.	Access'taki veri türleri ve alan uzunluklarından yararlanabilirsiniz.

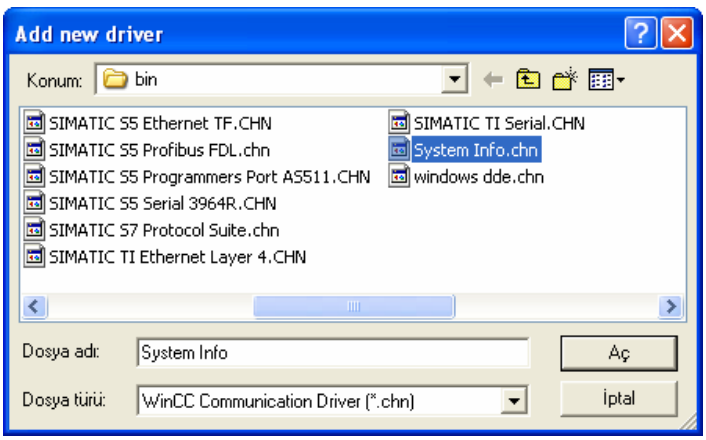
Veri tabanı Adı:									
Alan Adı		Alan Adı		Alan Adı		Alan Adı		Alan Adı	
Veri Türü	Alan Boyutu	Veri Türü	Alan Boyutu	Veri Türü	Alan Boyutu	Veri Türü	Alan Boyutu	Veri Türü	Alan Boyutu
Alan Adı		Alan Adı		Alan Adı		Alan Adı		Alan Adı	
Veri Türü	Alan Boyutu	Veri Türü	Alan Boyutu	Veri Türü	Alan Boyutu	Veri Türü	Alan Boyutu	Veri Türü	Alan Boyutu

ÖRNEK 5'te sadece okunan değer, Access tablosuna kaydedildi. Değerin kaydedilme tarihi ya da saati de kaydedilmek istenirse programda gereken düzenlemeyi yapınız.

Yol Gösterme:

strSQL = "INSERT INTO Tablo1 (Deger,Tarih) VALUES _
(“ & Zenta & "," “&Tarih&” ‘);"

Sistemi tarih ve saat tagleri system.ınfo.chn'den alınabilir.



ÖLÇME VE DEĞERLENDİRME

1. Aşağıdakilerden hangisi veri tabanı bileşeni **değildir?**

A) Tablolar
B) İndeksler
C) Satırlar
D) Bağlantı

Aşağıdaki cümlelerde boş bırakılan yere doğru kelimeleri yazınız.

2. Bir tablonun her satırında bir sütunun bir kez kullanılmasını işaret eden ……...'dır.

Aşağıdaki cümlelerde verilen bilgiler doğru ise (D), yanlış ise (Y) yazınız.

3. ( ) İlişkisel veri tabanının en önemli özelliği sorgulardan oluşmuş olmasıdır.

4. ( ) Ağ veri tabanı verilerini diske kaydedip yöneten program ilişkisel veri tabanımotorudur.

5. Aşağıdakilerden hangisi kullanıcıya bilgi sunmak için **yapılmamıştır?**

A) Formlar
B) Veri Erişim Sayfası
C) Raporlar
D) Modüller

6. Hangi nesne içine değer girişi **yapılamaz?**

A) Komut Düğmesi
B) Metin Kutusu
C) Liste Kutusu
D) Açılan Kutu

7. Aşağıdakilerden hangisi bir veri türü **değildir?**

A) Resim
B) Tarih/Saat
C) Sayı
D) Metin

DEĞERLENDİRME

Cevaplarınızı cevap anahtarıyla karşılaştırınız. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt ettiğiniz sorularla ilgili konuları faaliyete geri dönerek tekrarlayınız. Cevaplarınızın tümü doğru ise bir sonraki öğrenme faaliyetine geçiniz.

ÖĞRENME FAALİYETİ–2

AMAÇ

VB ile PLC'den alınan bir verinin MS Access tablosuna kaydedebileceksiniz.

ARAŞTIRMA

- Çok kullanıcılı veri tabanı uygulamalarını inceleyiniz.

2. VB İLE VERİ TABANI

Visual Basic ile veri tabanı uygulamaları yapmak için öncelikle veri tabanının oluşturulması gerekir. Microsoft Access, Dbase gibi veri tabanı programlarının oluşturduğu veri tabanlarının yanında VB ortamında da veri tabanı oluşturulabilir.

VB ortamında bu, iki yolla sağlanır:

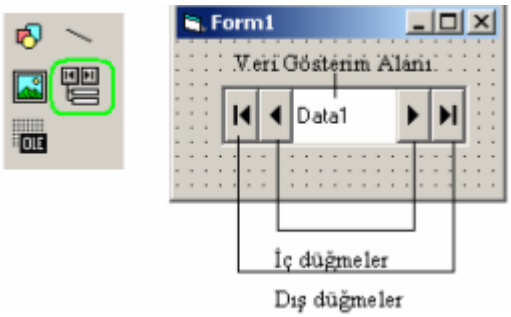
- Visual data manager(visdata) ile
- Kod yazarak

VisData'nın kullanımını sizlere bırakarak kod yazımı ile veri tabanı oluşturmaya bakılır.

2.1. Data Kontrol Bileşeni

VB'de daha önce hazırlanan veri tabanlarını incelemek ve üzerinde değişiklik yapmak için hazır bir bileşen vardır: "data kontrol bileşeni"

Visual Basic ile varolan bir veri tabanına erişiminin en kolay yolu, "data kontrol bileşeni"dir. Alet kutusu üzerinde bulunur. Form üzerine konulduğunda düğmeleri iç içe geçmiş gibidir.



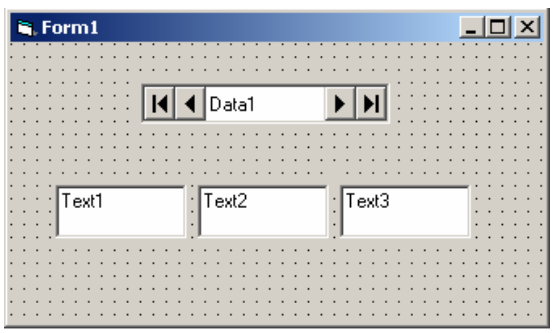
Şekil 2.1: Data kontrol

- İç düğmeler veri tabanı dosyasındaki kayıtlar arasında birer kayıt ileri ya da geri gezinmemizi sağlar.
- Dış düğmeler tablodaki ilk ve son kayıta gidilmesini sağlar.
- Tablodan görünmesi istenilen bilgi orta alanda görüntülenir.

Aşağıdaki örneklerde Visual Basic ile gelen bir diğer örnek veri tabanı dosyası olan Nwind.mdb kullanılacaktır. Nwind, Nortwind adındaki hayali bir şirketin demirbaş, müşteri, satıcı, çalışanlar, pazar istatistikleri vb. gibi tablolarını tutan bir veri tabanıdır.

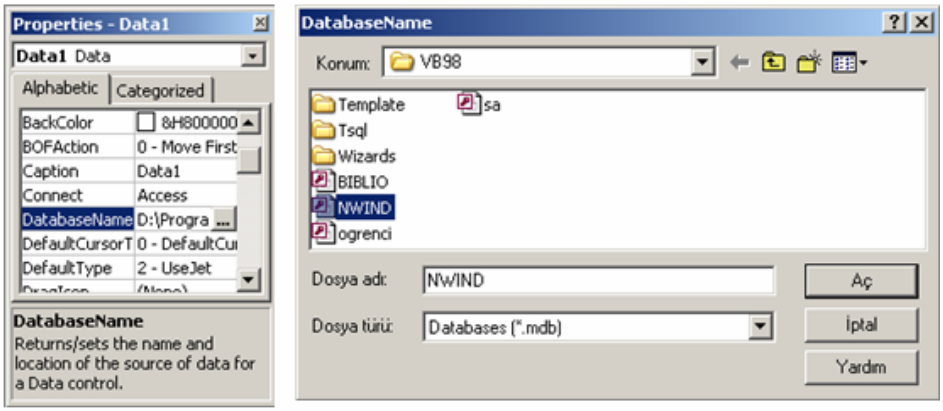
ÖRNEK 1: Data konrol bileşeninin basit kullanımı:

Form tasarımı aşağıda görüldüğü gibi yapılır.



Şekil 2.2: Data kontrol form tasarımı

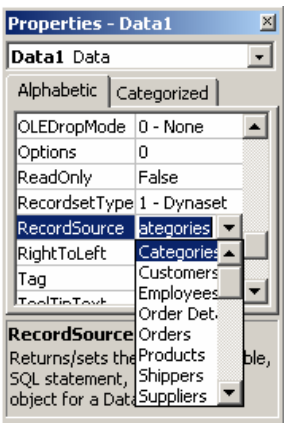
"Data control bileşeni"nin "databasename" özelliğine tıklanır. Karşımıza gelen ekrandan "Nwind dosyası" işaretlenir.



Şekil 2.3: Veri tabanı seçimi

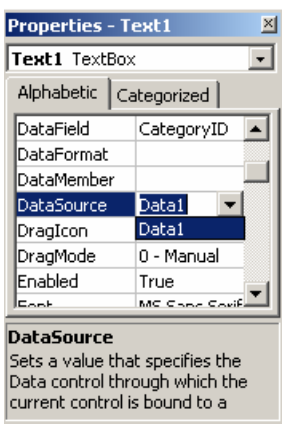
"Database" özelliği ayarlandıktan sonra "Visual Basic" veri tabanında bulunan tüm tablo ve sorgu isimlerini getirerek bunu "data control bileşeninin recordsource" özelliği olarak açılır-kapanır bir kutuya yerleştirmektedir.

"RecordSource" özelliğinde ise veri tabanı dosyası içinde bulunan tablolar listelenir. Burada görüntülenmek istenen tablo, ki burada "Categories" tablosu, seçilir.



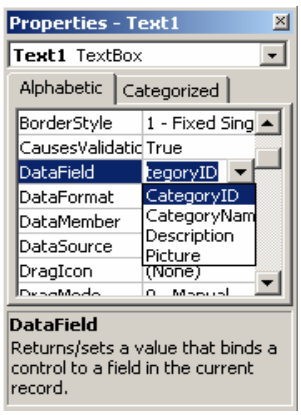
Şekil 2.4: Tablo seçimi

Metin kutularının "DataSource" özelliği, form üzerine konulan data object bileşeninin ismi olan "Data1"e eşitlenir.



Şekil 2.5: Veri kaynağı seçimi

Form üzerinde bulunan her bir metin kutusunun "DataField" özelliği, tablodaki bir alanla ilişkilendirilir.



Şekil 2.6: Alan seçimi

Form çalıştırıldığında ekran görüntüsü aşağıda görülmektedir. Düğmelere tıklandıkça kayıtlar arasında, kullanıcı gezinmektedir.

Şekil 2.7: Programın çalıştırılması

2.2. Recordset

Recordset kayıtların bir toplamıdır. Üç çeşiti vardır.

Data Control bileşeninin kullanıcının kayıtları silme ya da yeni kayıt eklemesine imkân vermediği dikkatlerden kaçmayacaktır. Kullanıcının bunları yapması istenirse kod yazmak gerekir.

Yazılan kodlar "Recordse" nesnesi etrafında toplanır. "Data kontrol bileşeninin DatabaseName ve RecordSource" özelliği ayarlandığında bir "RecordSet" nesnesi türetilir.

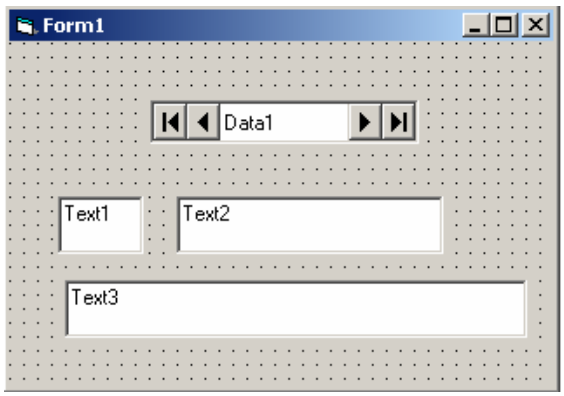
"Recordsetlerin" kendine has metotları vardır.

"RecordSet" dosya sistemlerinde olduğu gibi mevcut kayıtın konumunu takip etmek için bir işaretçi içerir. Kayıt işaretçisinin konumlandığı yerde hiçbir kayıt yok ise bir hata meydana gelir. "Recordset" bunu önlemek için BOF ve EOF olmak üzere iki özel konuma sahiptir.

BOF(beginning of file): Tablodaki ilk kayıttan önceki başlangıç konumu
EOF(end of file): Tablodaki son kayıtın bittiği konum

ÖRNEK 2: Kod ile data control bileşenini desteklemek

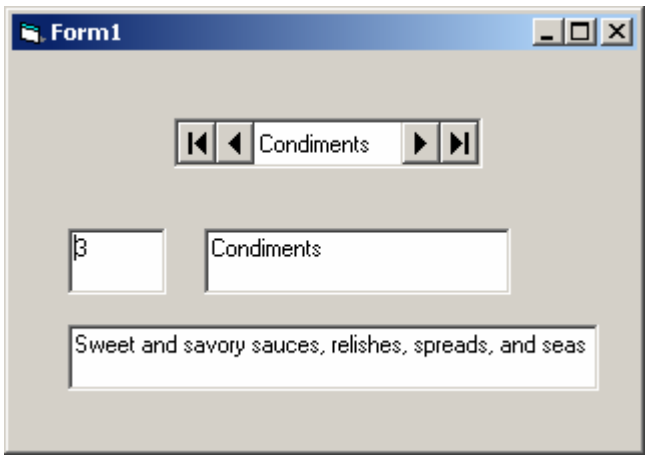
Aşağıdaki form tasarımını yaparak kodlar yazılabilir.



Şekil 2.8: Form tasarımı

```
Private Sub Form_Load()
   Data1.DatabaseName = "c:\NWind.mdb"
   Data1.RecordSource = "Categories"
   Data1.Refresh
   Text1.Text = Data1.Recordset.Fields("CategoryID").Value
   Text2.Text = Data1.Recordset.Fields("CategoryName").Value
     Text3.Text = Data1.Recordset.Fields("Description").Value
End Sub

Private Sub Data1_Reposition()
Text1.Text = Data1.Recordset.Fields("CategoryID").Value
Text2.Text = Data1.Recordset.Fields("CategoryName").Value
Text3.Text = Data1.Recordset.Fields("Description").Value
Data1.Caption = Data1.Recordset.Fields("CategoryName")
End Sub
```

Programı çalıştırılır.



Şekil 2.9: Programın çalıştırılması

'Recordset'lerin alan isimleri 'Field' özelliği ile metin kutularına yansıtılmıştır. Hangi alanın seçileceği parantez içinde yazılmış olup bunun değeri 'Value' özelliği aracılığıyla aktarılır.

```
Text1.Text = Data1.Recordset.Fields("CategoryID").Value
```

Program satırı

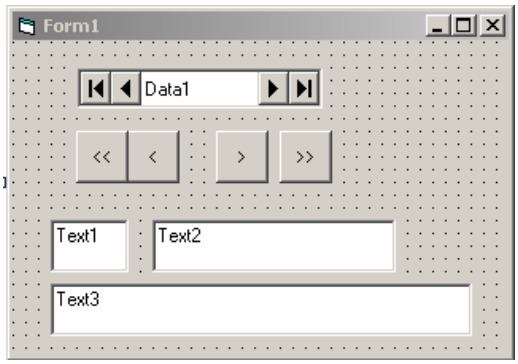
```
Text1.Text = Data1.Recordset.Fields("CategoryID")
```

şeklinde de yazılabilir. Böylelikle bir yazım kolaylığı sağlanmıştır.

Kayıtlar arasında dolaşabilmek için "Reposition olayı" kullanılır. 'Reposition olayı' kayıt işaretçisinin bir diğer kayıta konum değiştirdiğinde tetiklenir. Her yeni kayıtta değerler metin kutularına aktarılmaktadır. Ayrıca Data Control'ün 'Caption' özelliği de alanlardan birine eşitlenmiştir.

ÖRNEK 3: Reposition olayı

Aşağıda form tasarımı verilmiştir.



Şekil 2.9: Form tasarımı

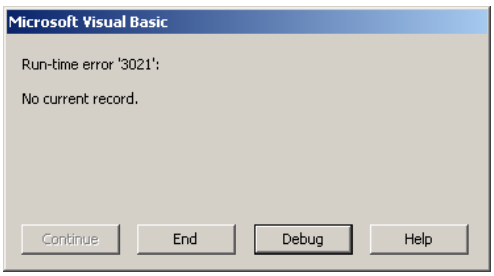
```
Private Sub Goster()
Text1.Text = Data1.Recordset.Fields("CategoryID")
Text2.Text = Data1.Recordset.Fields("CategoryName")
Text3.Text = Data1.Recordset.Fields("Description")
   Data1.Caption = Data1.Recordset.Fields("CategoryName")
End Sub

Private Sub Command1_Click()
Data1.Recordset.MoveFirst
                        Call Goster
End Sub

Private Sub Command2_Click()
Data1.Recordset.MovePrevious
Call Goster
End Sub

Private Sub Command3_Click()
Data1.Recordset.MoveNext
Call Goster
End Sub
```

Formun tasarım zamanında Data Control'ün 'Visible' özelliği 'False' yapılır. Burada yapılan Data Control'ün gezinti düğmelerinin yaptığı işi kendi atadığımız düğmelere yaptırmaktır.

Düğmelere sırasıyla

MoveFirst (İlk kayıta gider)
MovePrevious (Bir önceki kayıta gider)
MoveNext (Bir sonraki kayıta gider)
MoveLast (en son kayıta gider)

Metotları atanmış ve kayıtlar metin kutularında güncellenmiştir. Program çalıştırıldığında >> düğmeye tıklandığında program aşağıda görüldüğü gibi mevcut bir kaydın olmadığına dair bir hata mesajı verecektir.

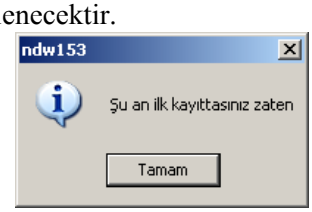


Şekil 2.10: Hata mesajı

Bunu önlemenin yolu daha önce anlatılan EOF ve BOF metotlarının kullanımıdır. Düğmelere ait program kodlarını aşağıdaki gibi değiştirelim.

```
Private Sub Command2_Click()
Data1.Recordset.MovePrevious
If Data1.Recordset.BOF Then
MsgBox "Şu an ilk kayıttasınız zaten", vbInformation, App.Title
Data1.Recordset.MoveFirst
End If
Call Goster
End Sub
Private Sub Command3_Click()
Data1.Recordset.MoveNext
If Data1.Recordset.EOF Then
MsgBox "Şu an SON kayıttasınız zaten", vbInformation, App.Title
Data1.Recordset.MoveLast
End If
Call Goster
End Sub
Private Sub Command4_Click()
Data1.Recordset.MoveLast
Call Goster
End Sub
```

Çalışma anında ilk kaydının öncesine gelindiğinde uyarı mesajı verilerek kayıt işaretçisi ilk kayıt üzerinde sabitlenecektir.



Şekil 2.11: Kendi hata mesajımız

2.3. ADO

ADO(avrivex data object), Microsoft'un veri tabanları ile çalışmak için geliştirdiği veri tabanı yönetim sistemidir. Bir anlamda ADO, veri tabanı nesnelerinin tasarım ve çalışma zamanındaki temsilidir ADO, mevcut olan eski DAO teknolojisinin bütünüyle yerine geçme iddiasında değildir. Aksine DAO modelini geliştirmiştir.

Microsoft'un e-mail hesaplarını, grafik formatlarını, internet kaynaklarını da kapsayacak şekilde geliştirdiği OLE DB teknolojisi üzerine kuruludur. ADO birbirinden çok farklı veri kaynaklarıyla uyumlu ve tutarlı çalışmak için bir arayüz sunar. Tüm veri kaynaklarına OLE DB kod seti ile bağlantı kurar. ADO, DAO'nun aksine yerel makinenin yanında ağ üzerindeki veri tabanlarını yönetir.

ADO Nesne Modeli:

ADO'nun yedi nesnesi vardır.

- Connection: Uygulama programı ile veri kaynağı arasındaki veri değişimini sağlar. Veri kaynağına erişim için bir veri kaynağı tedarikçisini(data provider) atar.
- Command: SELECT, UPDATE gibi bir SQL komutunu temsil eder.
- Recordset: Bir tablodan okunan veri Bu nesne içine kopyalanır. DAO'nun Recordset'i ve RDO'nun rdoResultset'i ile aynıdır.
- Field: Recordset'in her bir alanı
- Property: çeşitli ADO nesnelerinin özelliklerini işaret eder. Connection nesnesi SQL Server ya da Oracle'a bağlanırken oldukça farklı özellikler gösterir.
- Parameter: Command nesnesi sorgularda kullanılan parametrelere ihtiyaç duyabilir.
- Error: Program bir bağlantı kuramadığında, bir komutu işletemediğinde meydana gelir. Bu nesne, sonuç hatalarını tutar.

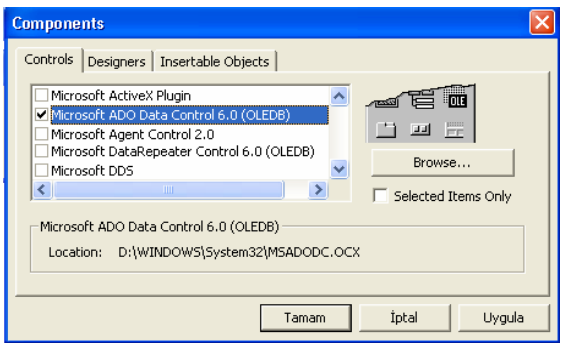
ADO nesne koleksiyon nesnelerini barındırır. Bir nesne koleksiyonu belirli tipte diğer nesne koleksiyonlarını içine alır. Connection koleksiyonu Errors koleksiyonuna, Recordset nesnesi Fields koleksiyonuna, Command nesnesi Parameters koleksiyonuna ve Connection, Command, Recordset ve Field nesneleri Properties koleksiyonuna sahiptir.

Her ne kadar nesnelerin silsileli bir yapısı olsa bile herhangi bir zamanda diğerinden bağımsız olarak oluşturabilir. Örneğin; Connection nesnesi olmadan Recordset oluşturuabilir. ADO'da DAO ve RDO'da olduğu gibi DBEngine ya da rdoEngine benzer bir yapılanma yoktur. ADO nesnelerinin diğerleri gibi mutlak bir sahibi yani en tepede bulunan bir nesne yoktur. Bu yüzden her bir connection nesnesi kendi Errors koleksiyonuna sahiptir.

ADO, hem görsel bir bileşen ile hem de kod yazımı ile program dâhilînde kullanılabilir.

İlk olarak kodsuz ADO aşağıda açıklanmıştır.

ADO bir ActiveX olduğundan programda kullanmadan önce alet kutusuna yüklenmesi gerekir. Bunun için Project menüsündan Components komutuna tıklanır ve Microsoft ADO Data Control 6.0 seçilir.



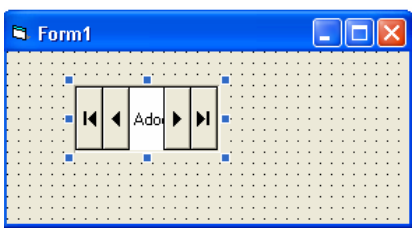
Şekil 2.12: ADO control

Sağ altta görülen simge ADO Control'un simgesidir.

Şekil 2.13: ADO control simgesi

ÖRNEK 4: ADO ile basit bağlantı

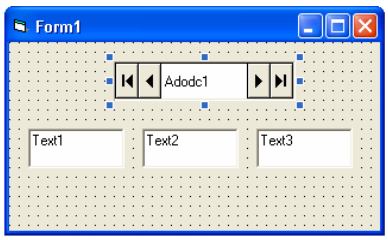
Form üzerine bir ADO çubuğu yerleştirilir.



Şekil 2.13: ADO nesnesi

ADO control çubuğu, görünüm olarak Data Control bileşeninden farklı değildir. Üzerindeki düğmelerin görevi aynıdır. Çalışma zamanında, ADO bir veri tabanına bağlı olduğunda düğmelerde kayıtlar arasında dolaşılabilir.

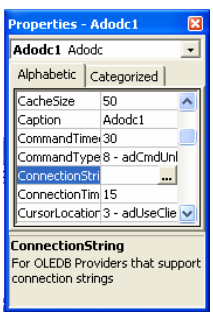
Üç tane metin kutusu form üzerine yerleştirilir.



Şekil 2.14: Form tasarımı

ADO DAO'da olduğu gibi doğrudan veri tabanı dosyasına bağlanan bir DatabaseName kullanmaz. Bunun yerine bilgisayar üzerinde bulunan bir ActiveX veri kaynağına kullanıcıyı bağlayan bir ConnectionString özelliğini kullanır. Bu esnada Data Control kullanılıyor olsaydı hemen DatabaseName özelliği, kayıtlı bir veri tabanına doğrudan irtibatlandırırdı. ConnectionString özelliği seçildiğinde ADO buna hazırlık kabilinden bir adım atar ve bir ActiveX veri nesnesi oluşturarak veri tabanı kayıtlarını tanımlamanızı bekler.

ADO'nun özellikler penceresine gidip sağındaki üç noktalı düğmeyi tıklanır.

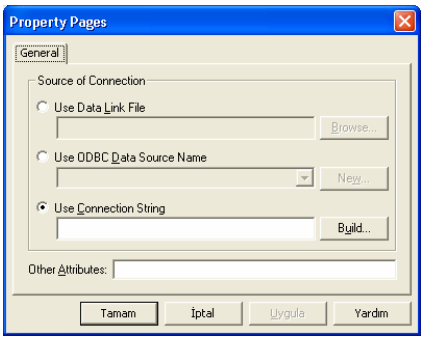


Şekil 2.15: Form tasarımı

ActiveX oluşturulduğunda karşıya Property Pages isimli pencerede üç seçenek çıkar.

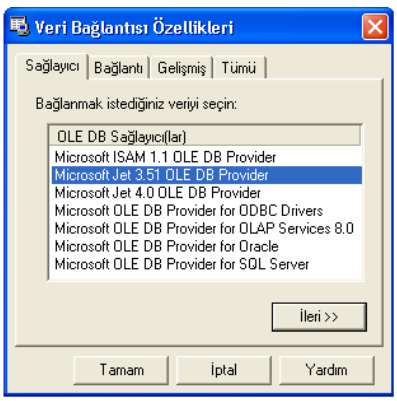
- OLE DB kısayol ismi
- ODBC Data Source Name (DSN: Veri kaynak ismi)
- OLE DB bağlantı dizisi

Burada connection string ile uğraşılacağından ilgili kutu işaretlenerek "Build" düğmesine tıklanır.



Şekil 2.16: Bağlantı cümlesi kurma

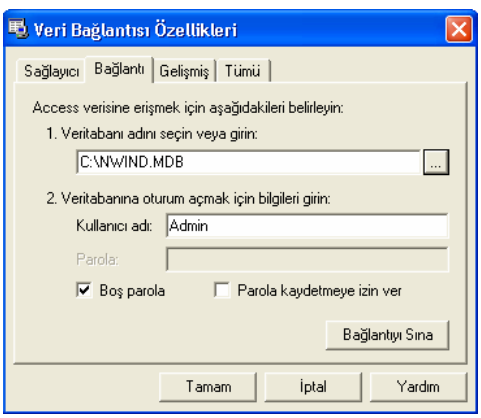
Servis sağlayıcı şekilde görüldüğü gibi seçilir.



Şekil 2.17: Veri bağlantı özellikleri

Burada seçilen OLE DB Provider, ADO'dan komutları alır ve veri kaynağının anlayacağı dile çevirir.

Program dâhilînde kullanılacak veri tabanı, karşımıza çıkan üç noktalı düğmeden yararlanarak seçilir.



Şekil 2.18. Veri bağlantı özellikleri

Tamam denildikten sonra bağlantı kurulacak veri tabanına nasıl ulaşılacağı Connectionstring ismine atanır.

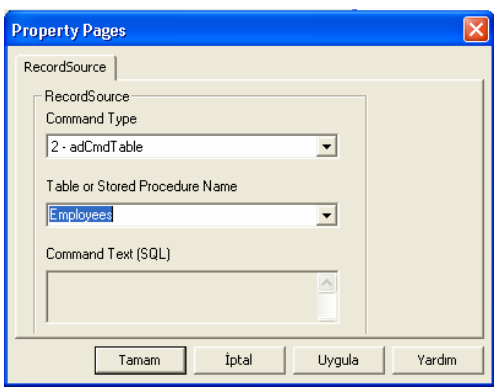
Provider=Microsoft.Jet.OLEDB.3.51;Persist Security Info=False;Data Source=C:\NWIND.MDB

Veri tabanı seçildikten sonra veri tabanından hangi tablonun kullanılacağı RecordSource özelliğine atanır.

Şekil 2.19: RecordSoruce özelliği

RecordSource'ın yanında görünen üç noktalı düğme tıklandığında özellikler penceresi açılır. Burada Command Type açılır kapanır kutusundan adCmdTable seçilir. Hemen aşağıdaki kutudan ise bir tablo işaretlenebir.

Burada bir tablo ile ilgilenileceğinden adCmdTable seçildi.

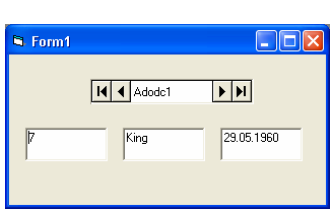


Şekil 2.20: RecordSoruce özelliği

Tamam dedikten sonra form üzerinde bulunan metin kutularının ADO ile bağlantısını yapmak için tabloda görülen özellik ayarlaması yapılır.

	DataSource	**DataField**
Text1	Adodc1	EmployeeID
Text2	Adodc1	LastName
Text3	Adodc1	BirthDate

Programı çalıştırıldığında aşağıdaki ekran görüntüsü alınır.

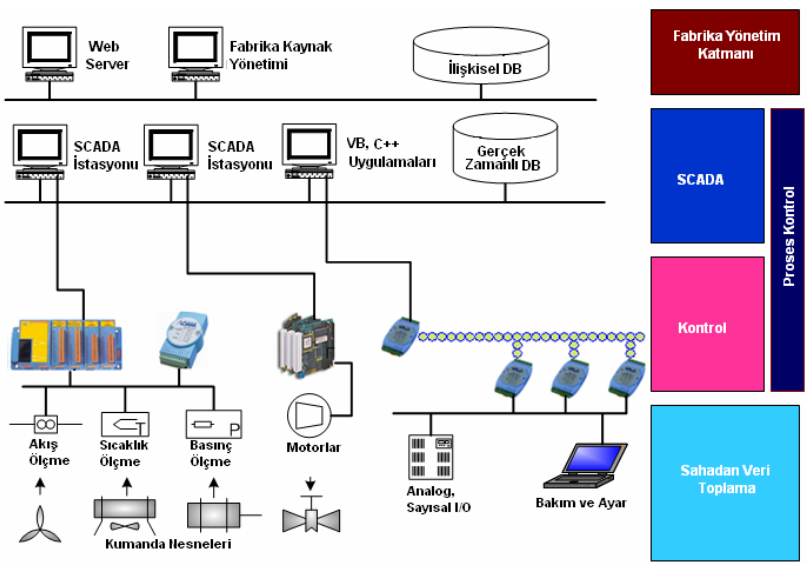


Şekil 2.21: RecordSoruce özelliği

2.4. OPC

Geleneksel kontrol sistemleri kapalı, kendine özgü haberleşme yapıları içermektedir. Bu tip kontrol sistemlerinde sahadaki verileri değişik firmaların ürünleri arasında dağıtmak zahmetli bir uğraştır.

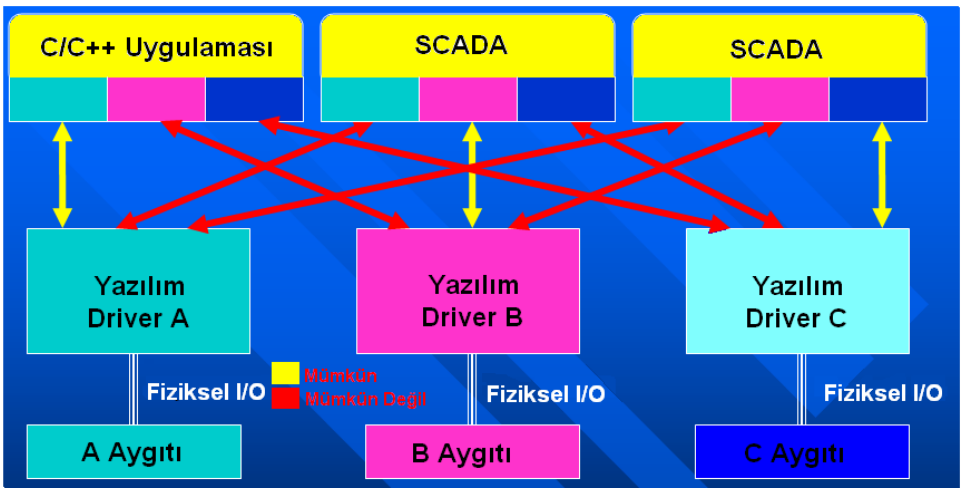
OPC standardı sayesinde, işletim sistemi farkı gözetmeksizin firma ağına bağlı herhangi bir bilgisayardan sahada bulunan ve bu standardı destekleyen herhangi bir donanımla haberleşmek son derece kolaylaşmaktadır. Özel donanım gerektirmeden, Ethernet alt yapısını kullanarak sahadan toplanan veriler bir OPC Server' dan okunabilir, bu veriler istenilen kaynağa iletilebilir. İster operatör ekranlarına, ister raporlama ve bakımda kullanılmak üzere bir veri tabanına ya da bir Excel dosyasına, ister cep telefonuna, hatta web ortamına aktararak istenilen her yere taşınabilir.



Şekil 2.22: Proses kontrol sisteminin mimarisi

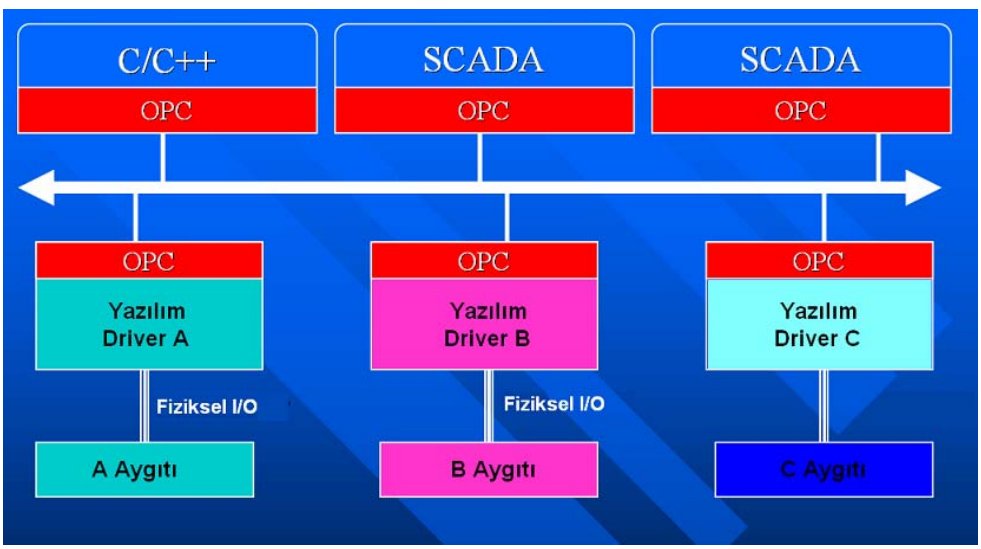
OPC kullanmayan sistemlerde:

- Her fiziksel eleman için ayrı driver gerekir.
- İki uygulama aynı anda bir sürücüye ulaşamaz.



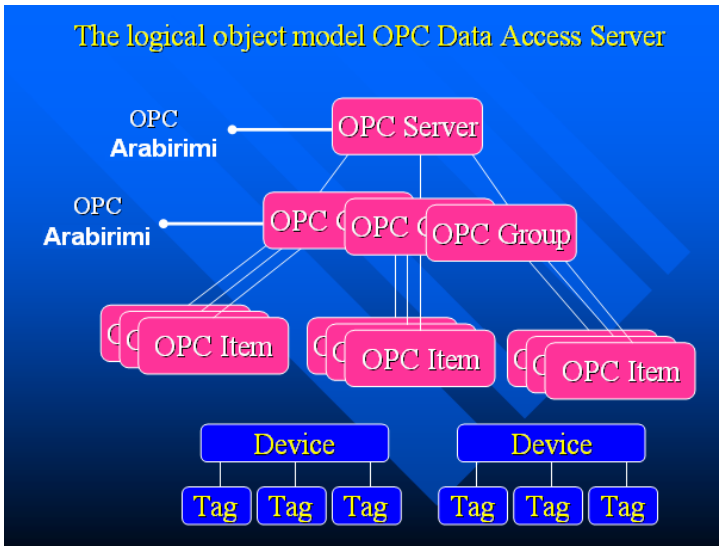
Şekil 2.23: I/O driver problemi

Şekil 2.5'te kırmızı hattan elemanlar SCADA ile haberleşemez. Farklı firmaların ürünleri arasında bir kaynaltırma yapan OPC'nin rolü Şekil 2.3'te görülmektedir.



Şekil 2.24: OPC ile problem çözümü

OPC ve SCADA'da PLCye bağlanan fiziksel elemanlar (I/O) Tag (etiket) olarak tanımlanır. Bazı yazılımlarda buna point(nokta) de denilmektedir. PLCden alınan bu değişkenler OPC içinde item(kalem) olarak tanımlanır. Aynı işi yapan kalemler bir zümre (grup) oluşturur. Zümreler OPC Server ile haberleşir.



Şekil 2.25: OPC ile sahadaki elemanların bağlanması

Piyasada pek çok OPC Server olmasına rağmen bu modülde iki tanesinin kullanımı açıklanmıştır.

- Omron CX OPC Server
- Kepware KEPServerEx

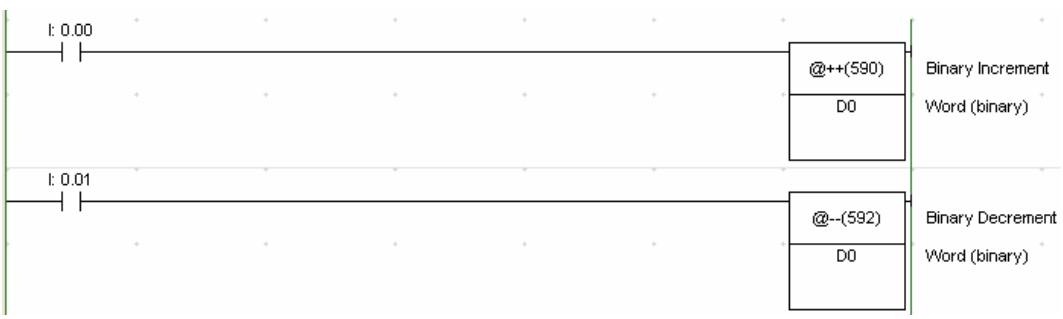
Her ikisinin de demosu internetten indirilebilir.

2.5. Omron CX OPC Server

Omron CX OPC Server bilgisayara kurulur.

ÖRNEK 5: Omron CJ1M PLCyi CX OPC Server üzerinden Visual Basic'e bağlanır.

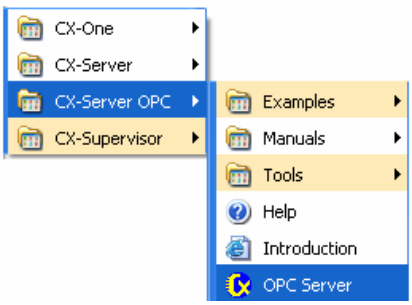
➢ CX Programmer'da aşağıdaki programı yazılarak Omron PLC'ye gönderilir.



Şekil 2.26: PLC Programı

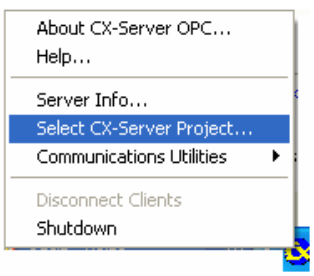
Burada 0.00 anahtarına basılınca D0 alanındaki sayı, bir artacak; 0.01 anahtarına basıldığında ise azalacaktır.

- OPC Server başlatılır.



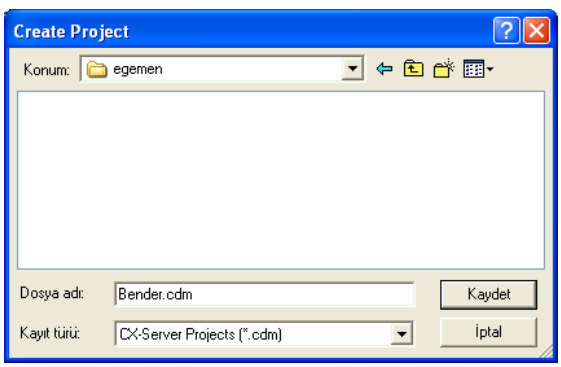
Şekil 2.27: Omron PLC OPC Server

- CX OPC, "system tray" adı verilen ekranın sağ alt köşesine yerleşir. Timsalin üzerinde sağ tuşla tıklanır.

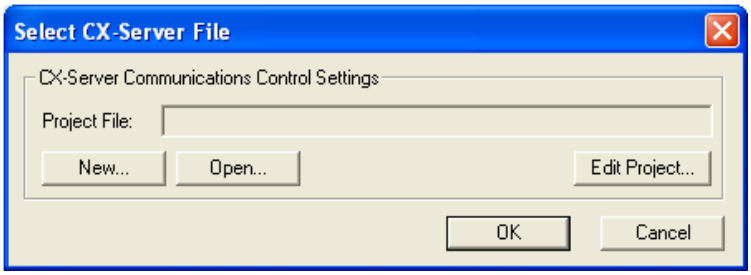


Şekil 2.28: Omron PLC OPC Server

- Açılan pencerede yeni bir proje ismi tanımlamak için "New" düğmesine basılır.

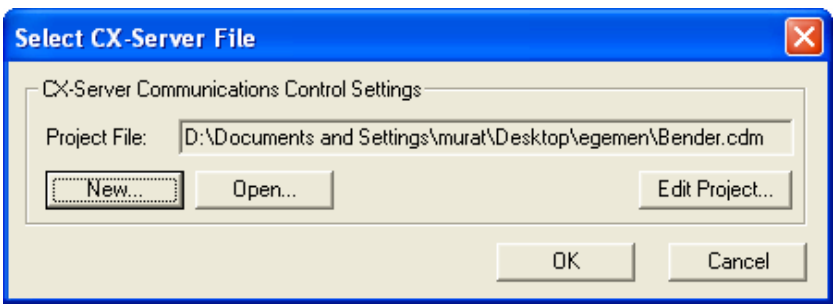


Şekil 2.29: Omron PLC OPC Server

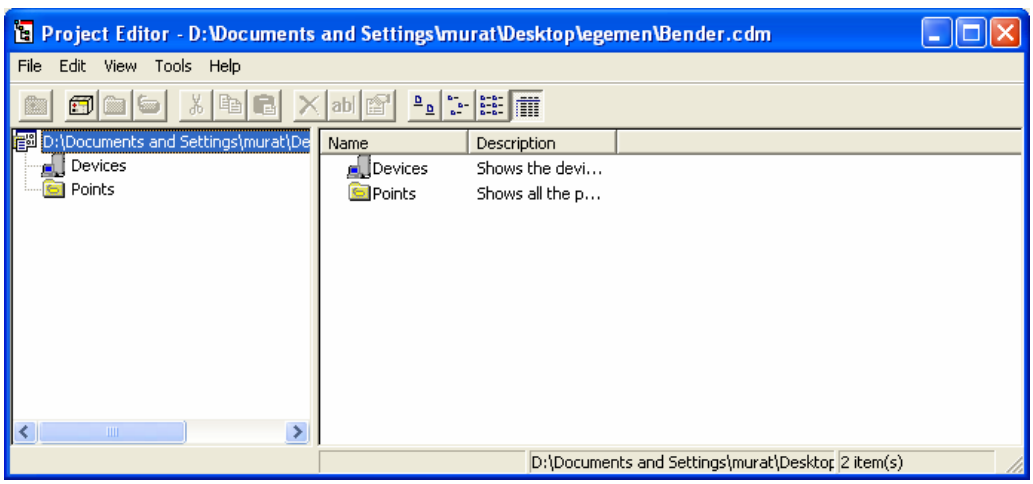


Şekil 2.30: CX Server dosyası

- Proje ismi yazıldıktan sonra "Edit Project" düğmesine tıklanarak PLC türü ve noktaların (point) girileceği "Project Editor" penceresi küşade edilir.

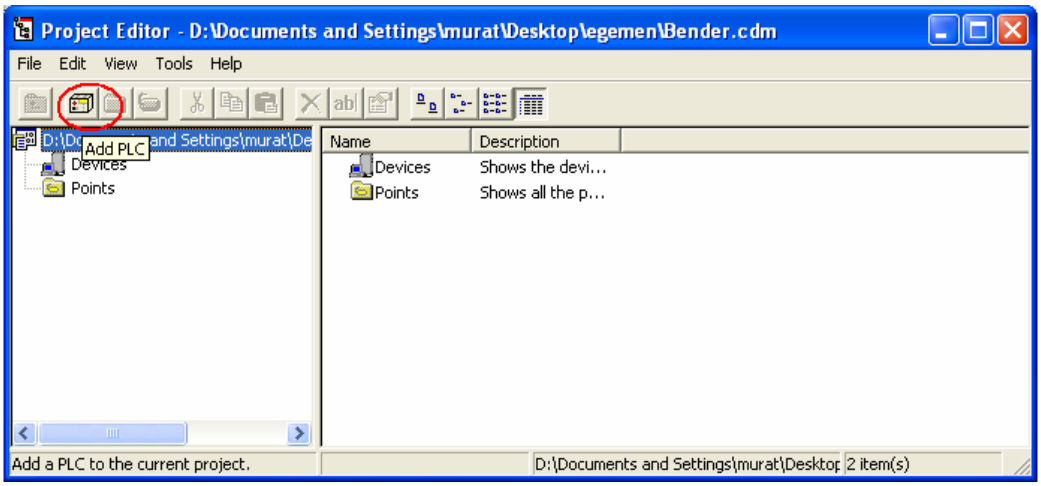


Şekil 2.31: CX Server dosyası



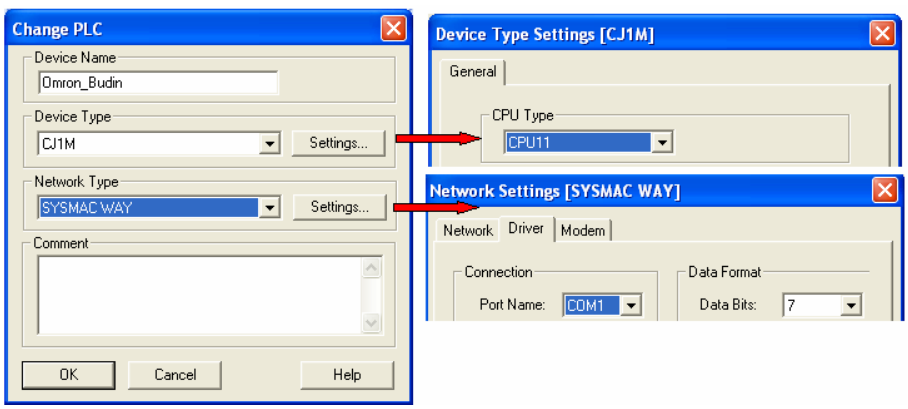
Şekil 2.32: CX Server dosyası

"Add PLC" düğmesine bir basılır.



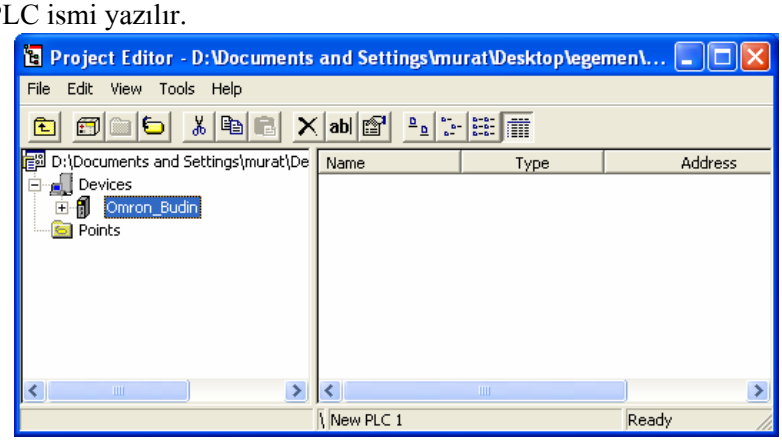
Şekil 2.33: OPC Server'e PLC ekleme

PLC ayar penceresinde bağlı olan PLC ayarları yapılır.



Şekil 2.34: OPC Server'e PLC ekleme

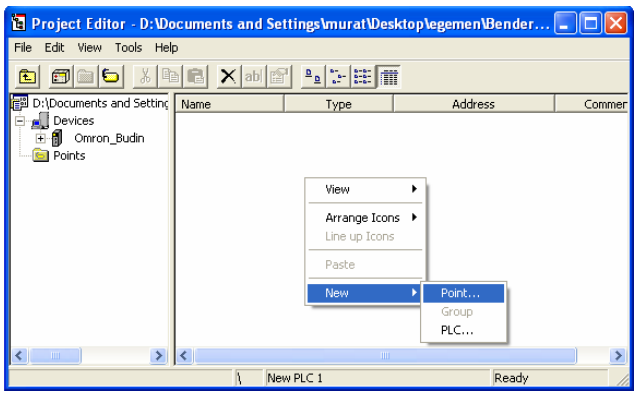
PLC ayarı yapıldıktan sonra proje ağacında bulunan Devices alanının altına tanımlanan PLC ismi yazılır.



Şekil 2.35: OPC Servere PLC Ekleme

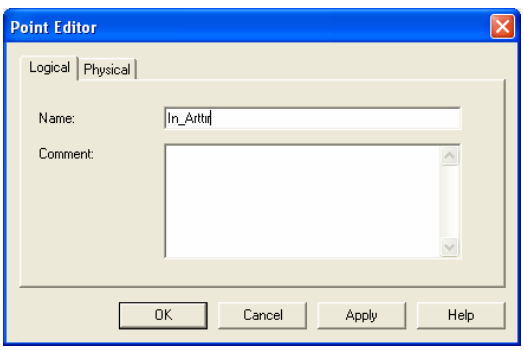
Bu esnada PLC ayarları değiştirilmek istenirse PLC ismi üzerinde sağ tuş tıklanarak "Properties"'den gerekli düzeltmeler yapılabilir.

- Sonra PLC'de giriş-çıkış ya da hafıza bölgelerinin temsil eden noktaları tanımlanmalıdır. Bunun için pencerenin sağ tarafındaki ıssız alanda sağ tuşa basılarak "point" seçilir.



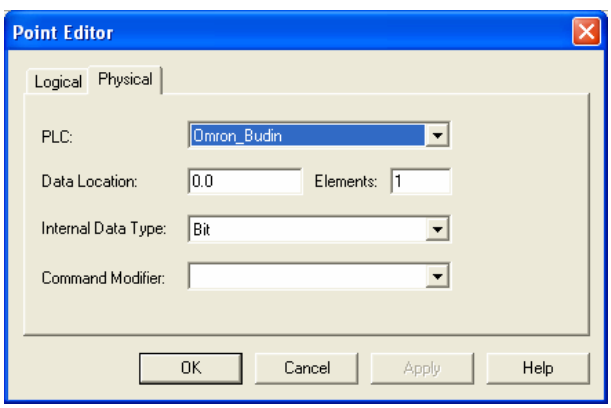
Şekil 2.36: Point ekleme

PLC programında 0.00 nolu girişi temsil edecek nokta ismi yazılarak "Physical" sekmesine basılır.



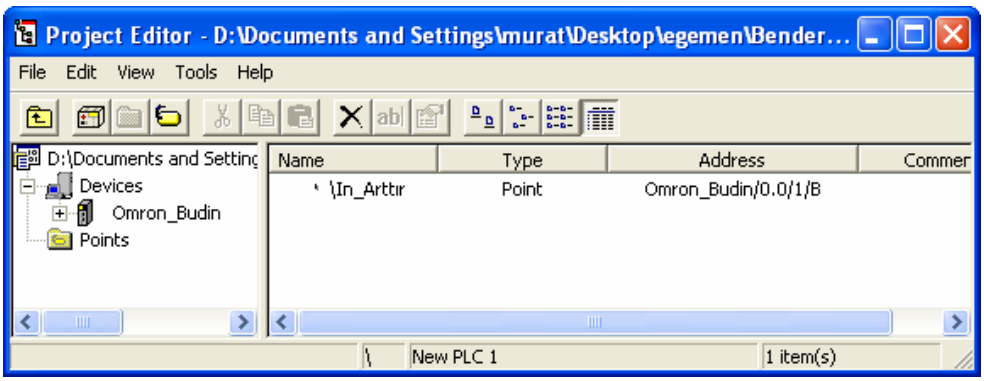
Şekil 2.37: Point ekleme

Burada da tanımlamalar yapılır.



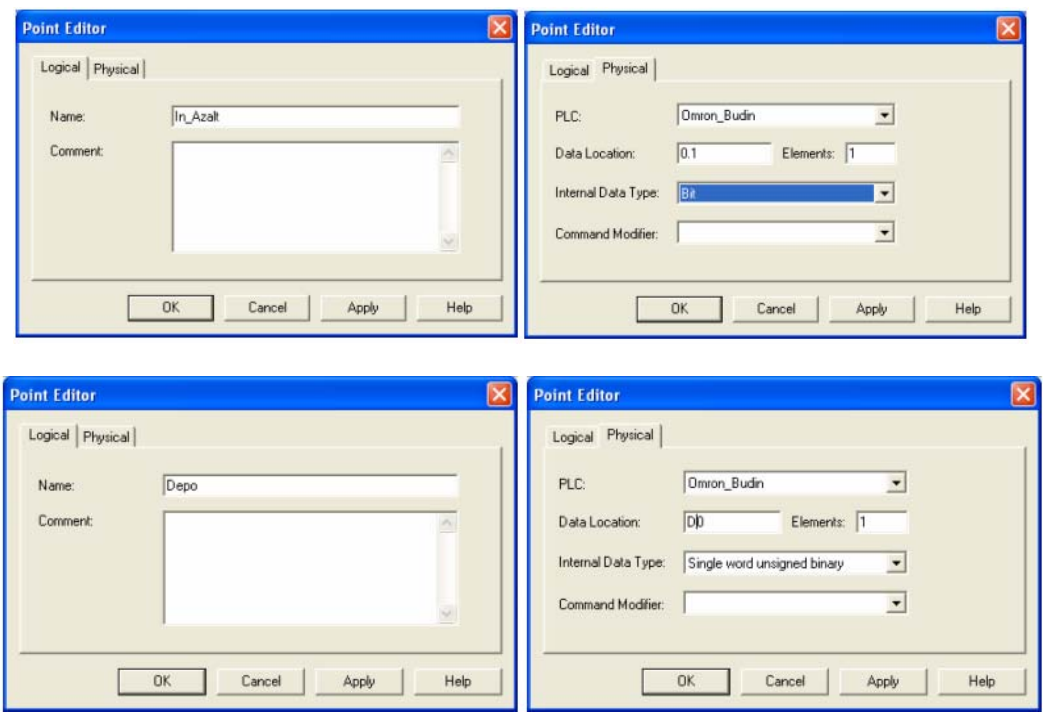
Şekil 2.38: Point ekleme

Girilen nokta ismi proje penceresinde görülecektir.



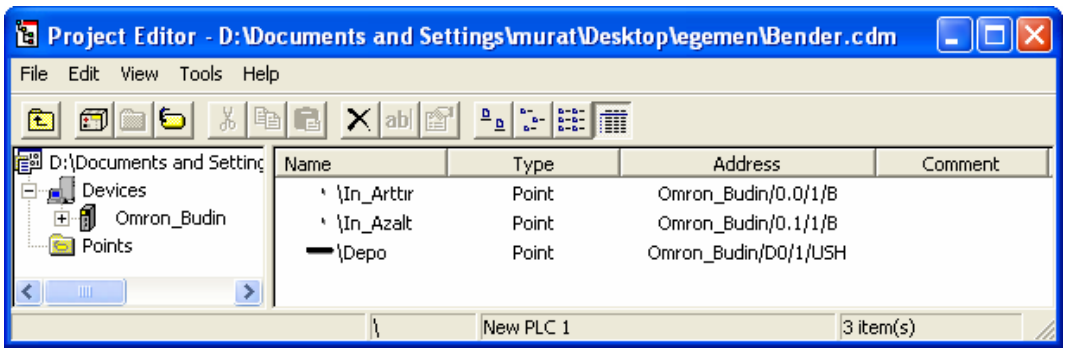
Şekil 2.39: Point ekleme

PLC programındaki 0.01 girişini ve D0 alanı temsil eden tanımlamalar yapılır.



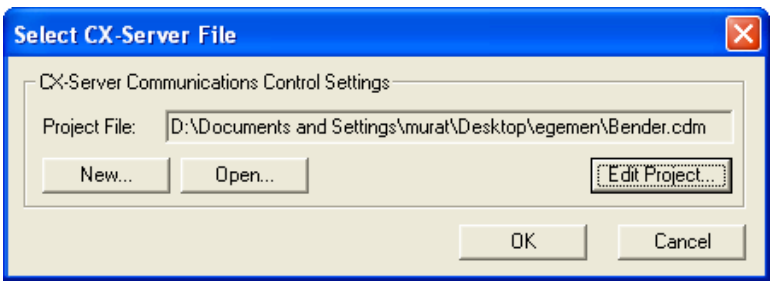
Şekil 2.40: Point ekleme

Girilen tüm noktalar proje penceresinde görülmektedir.



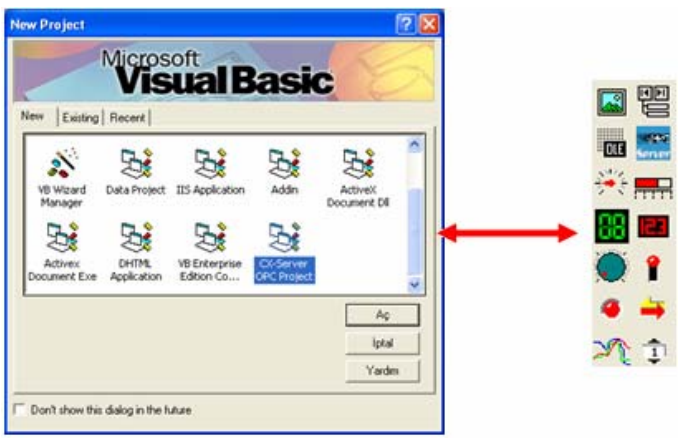
Şekil 2.41: Point ekleme

- Tanımlamalar bittikten sonra önce "Project Editor", ardından CX Server File penceresinde OK tuşuna basılarak pencereler kapatılır.



Şekil 2.42: OPC Server dosyası

- VB 6.0 açılır. New Project penceresinde "CX-Server OPC Project" seçilir. CX Server OPC programı VB kurulduktan sonra bilgisayara yüklenirse bu şablon kendiliğinden gelir. CX-Server'ın hazır Active X kontrolleri alet kutusuna otomatik olarak yüklenecektir.

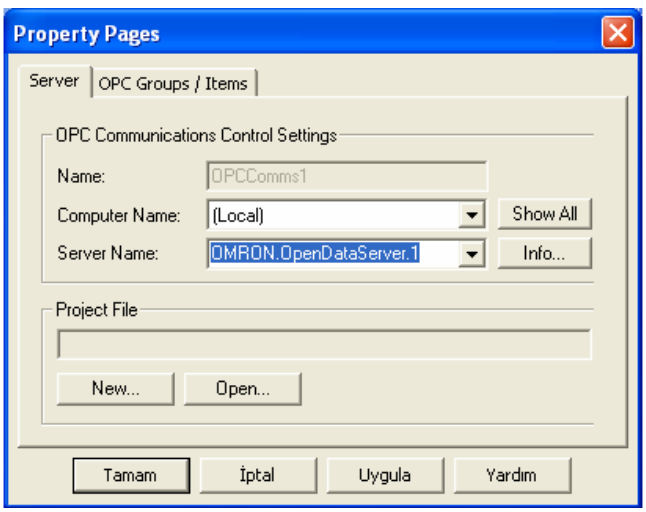


Şekil 2.43: Visual basic'te yeni proje oluşturma

➢ Form üzerine "OPC Communications Control" nesnesi yerleştirilir. Bu kontrol CX Server uygulaması(VB, Excel) ile OPC Server arasındaki arabirimi tesis eder. Program tasarım modunda iken bu nesne görünür. Çalışma anında saklanır.

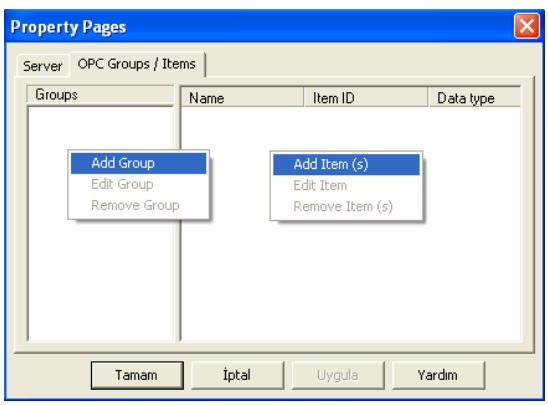
Şekil 2.44: Visual basic'te yeni proje oluşturma

Nesne, Form üzerine yerleştikten sonra üzerinde sağ tuşla tıklanarak "Properties" seçilir.



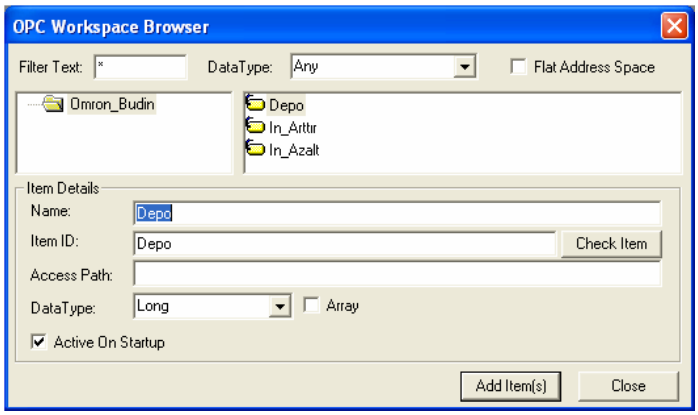
Şekil 2.45: OPC Server özellikler penceresi

Server Name olarak "Omron.OpnDataServer.1" seçilir. New düğmesine tıklanarak proje ismi verilir. "OPC Groups/Items" sekmesine tıklanarak ekranda zuhur eden "Property Pages" penceresinde Groups alanında sağ tuşla tıklanır ve "Add Group"'tan yeni bir zümre ismi girilir.



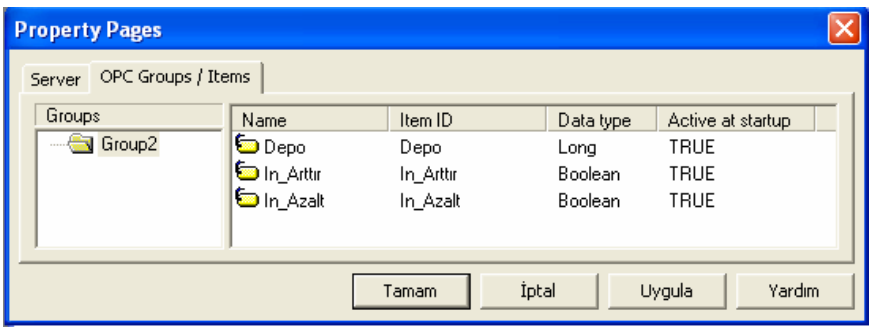
Şekil 2.46: Point seçimi

Zümre ismi belirlendikten sonra sağ taraftaki boş alanda yine sağ tuştan yardım alarak "Add Item" (Parça-kalem ekle) seçilir. Bu esnada Omron OPC açık olmasa bile "OPC Communications Control" nesnesi tarafından açıldığı görülür.



Şekil 2.47: Point seçimi

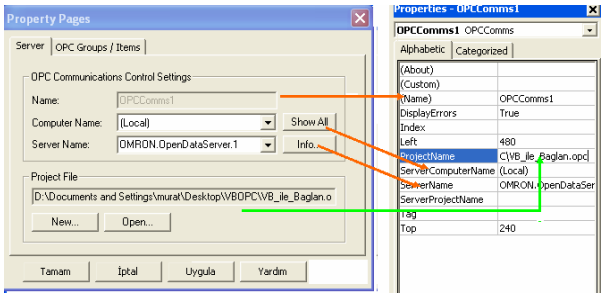
Daha önce OPC'de tanımlanan tüm 'tag'lar görünür. Her birini seçerek "Add Items(s)" düğmesi ile sırayla seçilerek Property Pages penceresine nakliyatı sağlanır. Seçim bittikten sonra pencere kapanır.



Şekil 2.48: Point seçimi

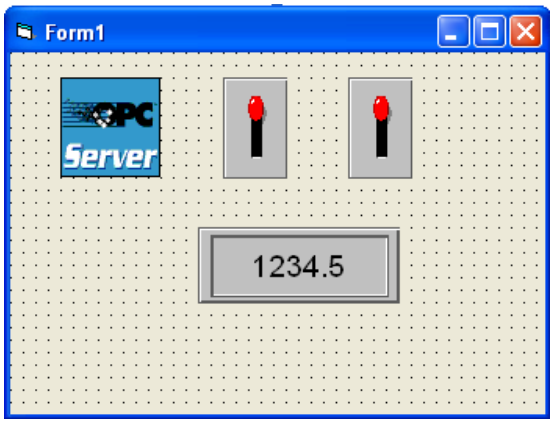
Tamam denilerek pencere kapatılır ve VB formuna geri dönülür. OPC ile VB'nin irtibatı sağlanmış olunur.

Ayarlar bittikten sonra "OPC Communications Control" seçili iken Properties tablosuna bakılırsa ilgili satırların doldurulduğu görülür.



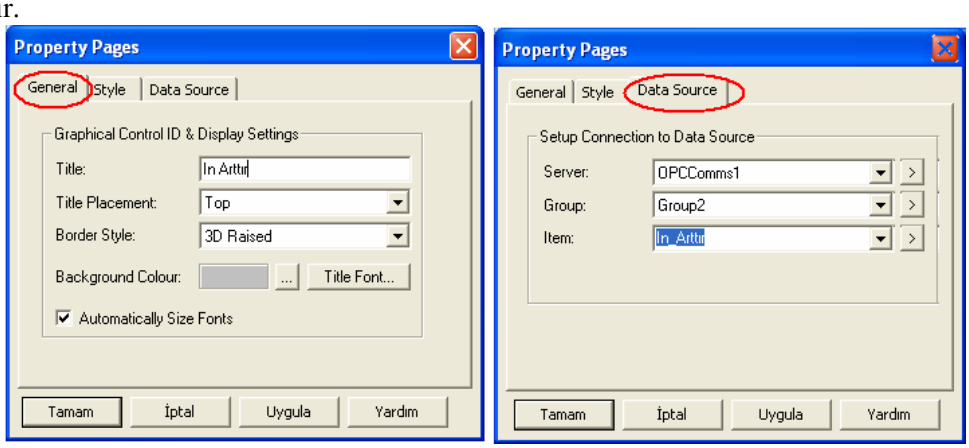
Şekil 2.49: Point özellikleri

- Üç değişkeni form üzerinde gösterecek nesnelerin eklenmesine geçilir. Bunun için 2 tane Toggle butonu bir tane de Display bileşeni yerleştirilir.

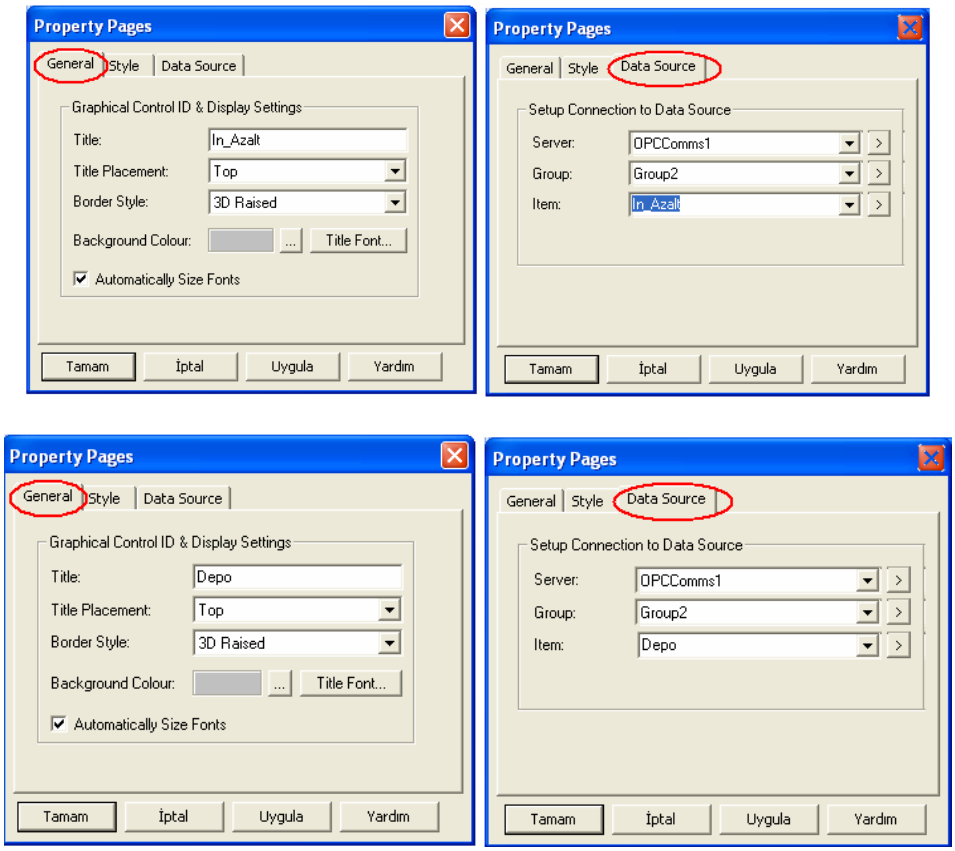


Şekil 2.50: Form tasarımı

Toggle butonunda sağ tuşa tıklayarak "Properties" seçilirek aşağıdaki ayarlamalar yapılır.

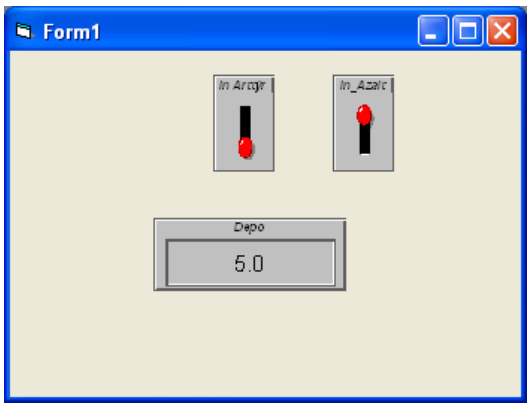


Şekil 2.51: Nesneye point bağlanması



Şekil 2.52: Nesneye point bağlanması

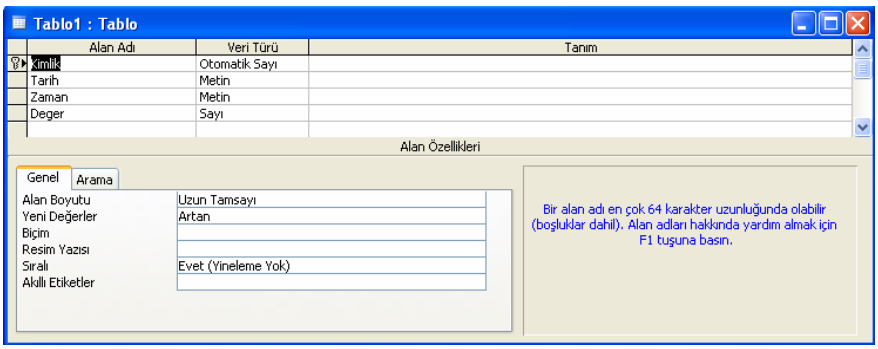
- Program hazır hâle gelir. Program çalıştırılır.



Şekil 2.53: Programın çalıştırılması

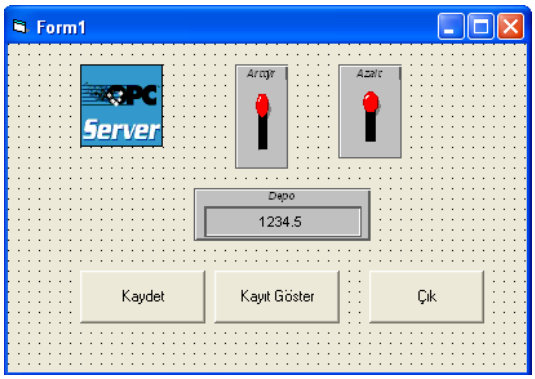
ÖRNEK 6: Elde eldilen Depo bilgisini MS Access veri tabanına kaydetme

Bunun için MS Access'te görüldüğü gibi bir tablo oluşturulur. Acess 97 formatında kaydedilir. VB'deki Data Control bileşeni bu formatı desteklemektedir.



Şekil 2.54: Acces de Tablo Oluşturma

➢ Form tasarımı geliştirilebilir.



Şekil 2.55: Form tasarımı

Projeye ana formun yanına 2 form ve bir modül ilave edilir.

Ana formun kod kısmı:

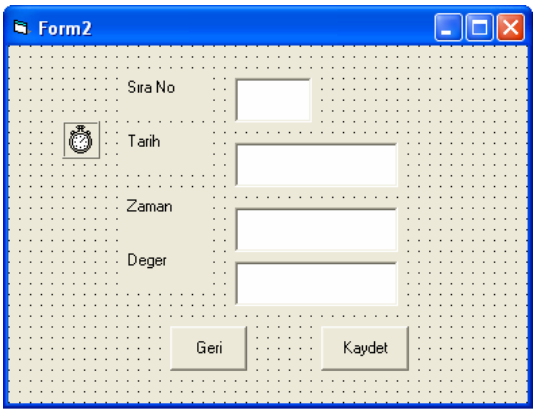
```
Private Sub cmdGoster_Click()
Form3.Show
End Sub

                    Private Sub cmdKaydet_Click()
Form2.Show
End Sub

Private Sub cmdTerk_Click()
End
End Sub

    Private Sub Form_Load()
    Data1.DatabaseName = "c:\NWind.mdb"
    Data1.RecordSource = "Categories"
    Data1.Refresh
    Text1.Text = Data1.Recordset.Fields("CategoryID")
    Text2.Text = Data1.Recordset.Fields("CategoryName")
    Text3.Text = Data1.Recordset.Fields("Description")

    End Sub
```

Modül kısmı:

```
Public mDb As Database
                Public mRs As Recordset

Global Depo_Deger As Integer
```

DAO modelini kullanmak için referansta bulunmak gerekir.

2. formun tasarımı:

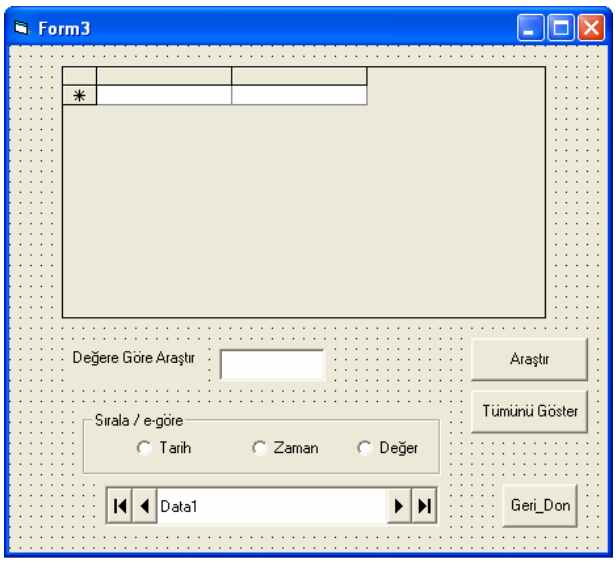


Şekil 2.56: Form tasarımı

Formun kodu:

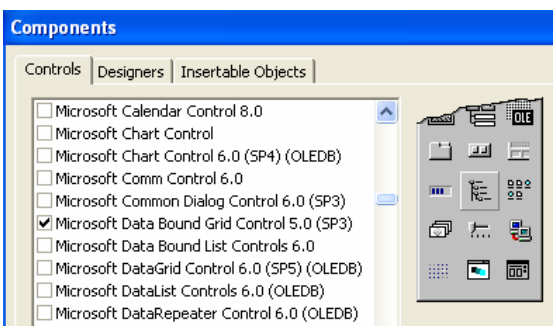
```
Dim MyTime, MyDate, MyStr

Private Sub cmdGeri_Click()
Unload Me
End Sub
Private Sub cmdKayıt_Click()
mRs.AddNew
mRs!Tarih = Text2.Text
mRs!Zaman = Text3.Text
mRs!Deger = Text4.Text
mRs.Update
End Sub
Private Sub Form_Load()
Set mDb = OpenDatabase(App.Path & "\budin.mdb")
Set mRs = mDb.OpenRecordset("Tablo1", dbOpenTable)
Text1.Text = "Otomatik sayı"
Text1.Enabled = False
                    End Sub
Private Sub Form_Unload(Cancel As Integer)
   mRs.Close
   mDb.Close
   Set mRs = Nothing
   Set mDb = Nothing
End Sub

Private Sub Timer1_Timer()
MyTime = Format(Time, "h:m:s")
MyStr = Format(Date, "Long Date")
MyDate = Format(MyStr, "dddd, mmm d yyyy")
Text2.Text = MyDate
Text3.Text = MyTime
Depo_Deger      =      Form1.OPCComms1.Value("Group1",
"Depo")
```

Sonuncu formun tasarımı:



Şekil 2.57: Form tasarımı

Burada DBGrid kullanılmıştır. Projects/Components'den seçilmelidir.



Şekil 2.58: Dbgrid nesnesi

Option butonlar dizili(array) tanımlanmalıdır.

DataControl bileşenini ayarları:

- Name: Data1
- Connect=Access; DatabaseName=budin.mdb
- RecordsetType=1-DynaSet
- RecordSource=Tablo1

DB Grid için:

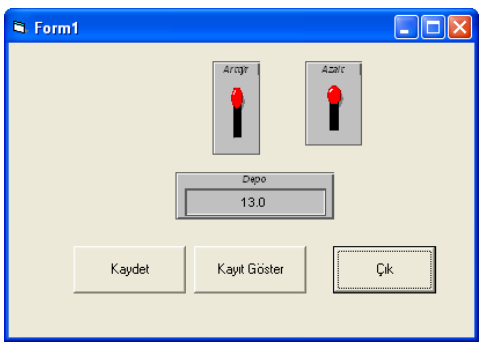
DataSource=Data1

Formun kodları:

```
Private Sub cmcArastir_Click()

                    Dim strSql As String

On Error GoTo ErrHandler

strSql = "SELECT * FROM Table1 "
Text1.Text = Trim(Text1.Text)
If (Text1.Text <> "") Then
   strSql = strSql & "WHERE Deger LIKE '" & Text1.Text & "*'"
End If

Data1.RecordSource = strSql
Data1.Refresh
Data1.Recordset.MoveLast
Data1.Recordset.MoveFirst
Exit Sub

ErrHandler:
   MsgBox "Kıstasınıza uyan bir kayıt bulunamadı"
   Reset_RecordSet
End Sub

Private Sub Reset_RecordSet()
Dim strSql As String

strSql = "SELECT * FROM Tablo1 "
Refresh_RecordSet strSql
End Sub

Private Sub cmdGeri_Click()
Unload Me
End Sub
Private Sub Form_Activate()
Data1.Recordset.MoveLast
Data1.Recordset.MoveFirst
End Sub
```

```
Private Sub Data1_Reposition()
Data1.Caption = "Depo Değeri: " & Data1.Recordset.RecordCount & " / "
& (Data1.Recordset.AbsolutePosition + 1)
End Sub

Private Sub optSort_Click(Index As Integer)
Dim strSql As String

strSql = "SELECT * FROM Tablo1    "

If (Index = 0) Then
 strSql = strSql & " ORDER BY Tarih"

ElseIf (Index = 1) Then
 strSql = strSql & " ORDER BY Zaman"

Else
 strSql = strSql & " ORDER BY Deger"

End If
Refresh_RecordSet strSql
End Sub

Private Sub Refresh_RecordSet(strSql As String)
Data1.RecordSource = strSql
Data1.Refresh
Data1.Recordset.MoveLast
Data1.Recordset.MoveFirst
End Sub

Private Sub cmdTum_Click()
Reset_RecordSet
End Sub
```

- Programı çalıştırılır.



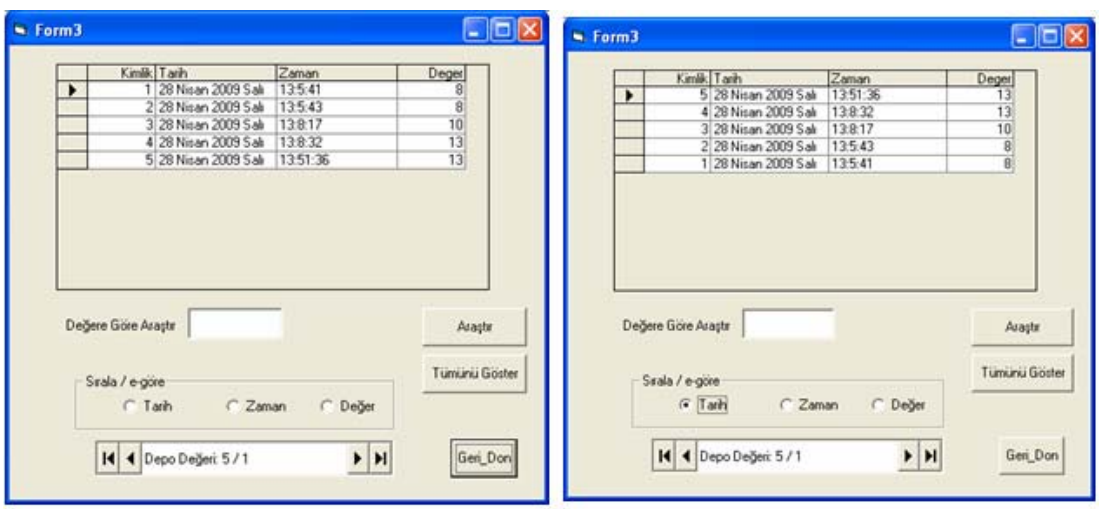
Şekil 2.59: Programın çalıştırılması

Kaydet düğmesine basılır.

Şekil 2.60: Kaydet düğmesine basıldığında

PLC'den veri girişi oldukça 'Kaydet' düğmesine basıldığında veri tabanına kaydedecektir. Kayıttan sonra ana forma geri dönülebilir.

Kayıt Göster düğmesine basılır.



Şekil 2.61: Kaydet düğmesine basıldığında

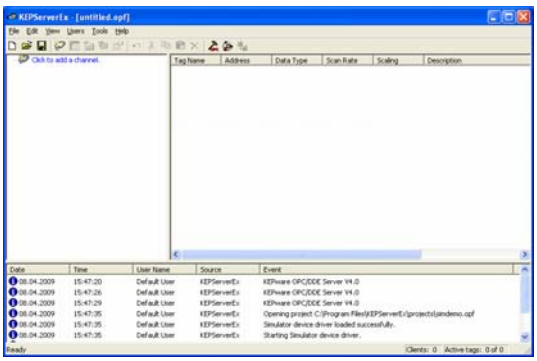
2.6. Kepware OPC

Kepware OPC bilgisayara kurulur.

ÖRNEK 7: Kepware ile VB üzerinde veri tabanı

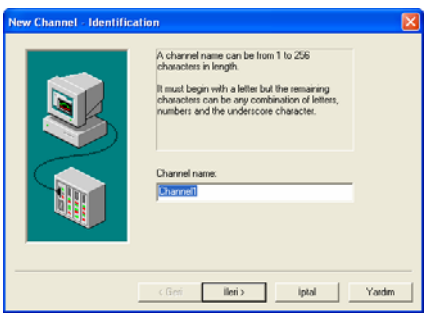
Bir önceki örneği, bir diğer OPC Server olan Kepware ile yapalım. PLC'deki program değişmeyecek.

KEPServerEx programını kurduktan sonra çalıştırılır.



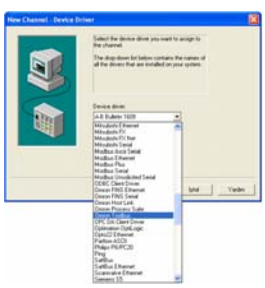
Şekil 2.62: Kepware OPC server programı

Sol üst köşede bulunan "Click to add channel" üzerine çift tıklayarak Omron PLC'yi tanıtılır. PLC marka ismi olarak bir isim girilebilir.



Şekil 2.63: Yeni kanal ekleme

Omron CJ1M üzerinde iki tip programlama portu var. Eğer peripheral portu kullanılacaksa Omron Toolbus, COM1 kullanılacaksa Omron FINS Serial seçilir.



Şekil 2.63: Bağlantı elemanlarını tanımlama

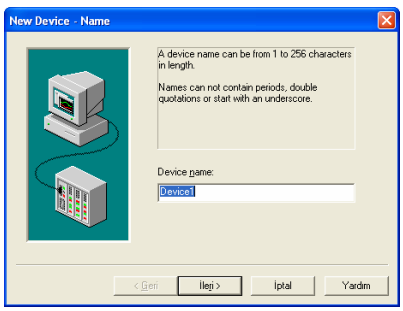
Diğer ayarlar da değiştirilebilir.

Şekil 2.64:Bağlantı ile ilgili diğer parametreler

PLC markası girildikten sonra türü girilmelidir. Channel1'in altında "click to add a device" çift tıklanır.

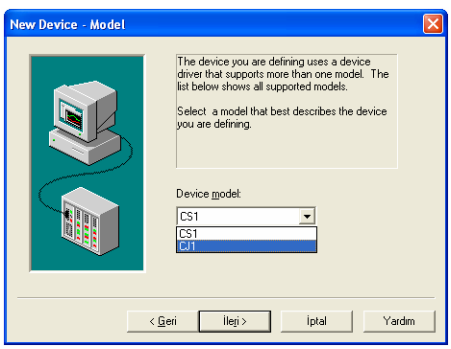
Şekil 2.65:Kanal tanımlanmış durumda

Bir isim ataması yapılır.



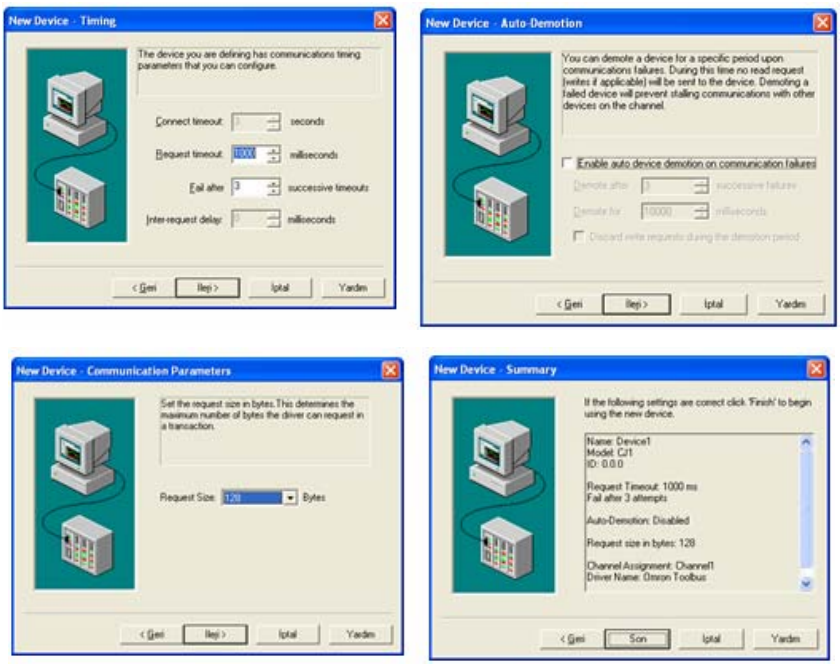
Şekil 2.65:Kanala bağlı cihaz tanımlama

Tip olarak CJ1M seçilir.



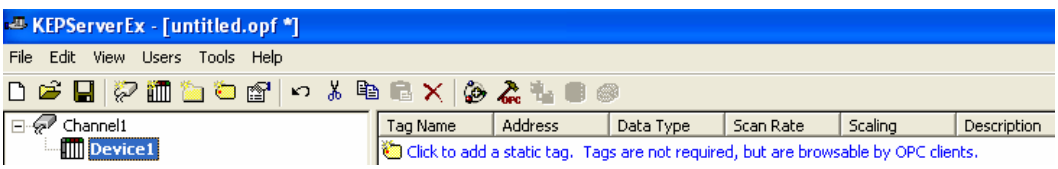
Şekil 2.66:PLC tipini belirleme

Diğer ayarlar değiştirilmeden devam edilebilir.



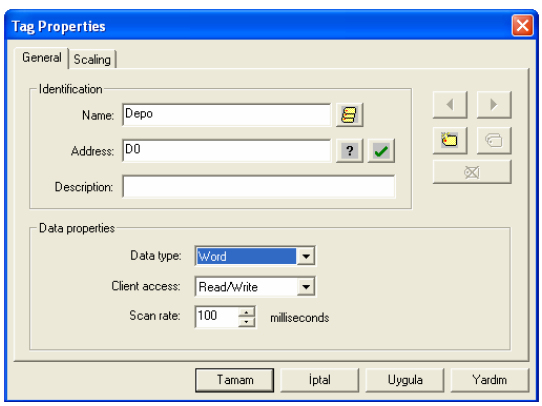
Şekil 2.67:Diğer parametreler

Sağ taraftaki boş alanda sağ tuşa tıklayarak "New Tag"'dan tag eklenmesi gerekir.



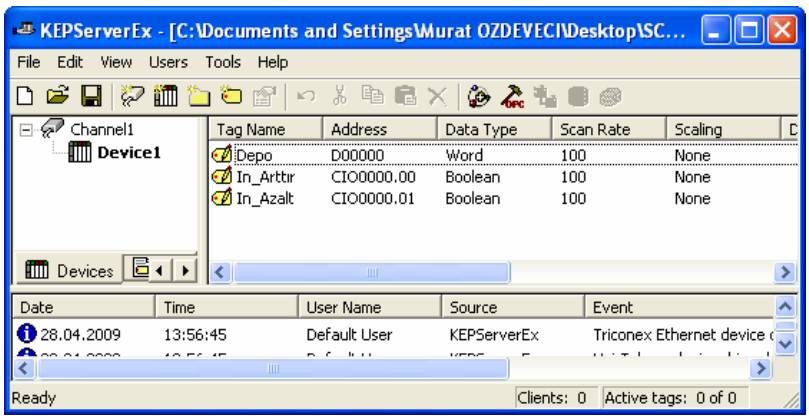
Şekil 2.68:"Tag" yeni etiket tanımlama

Önceki örneklerde bahsi geçen üç değişken tanımlanabilir.



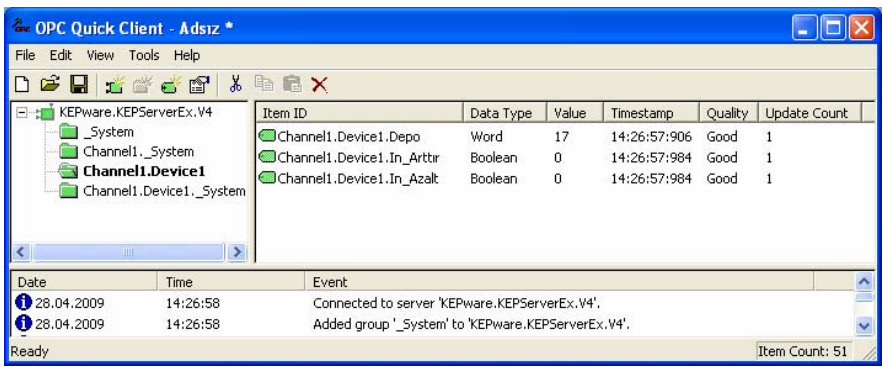
Şekil 2.69:"Tag" yeni etiket tanımlama

Diğer etiketler de tanımlanır.



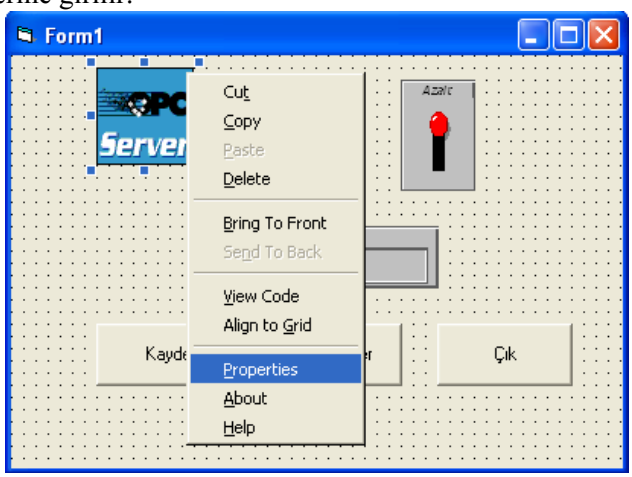
Şekil 2.70: Etiketler

Bu esnada üzerinde çekiç resmi olan OPC düğmesine tıklanırsa çalışma zamanında (real time) PLC ile veri alışverişi yapılabilir.



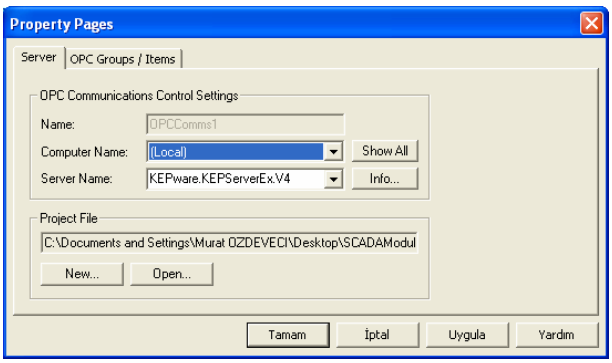
Şekil 2.71: Gerçek zamanlı PLC iletişimi

Visual Basic'te form tasarımı ve kodlar hemen hemen aynı olacaktır. OPCComms bileşeninin özelliklerine girilir.



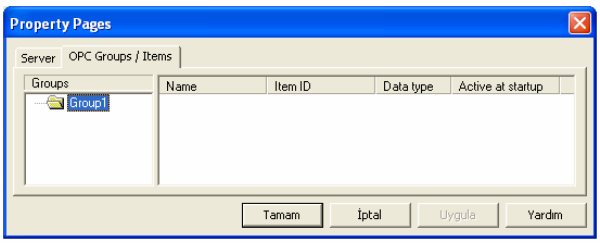
Şekil 2.72: Gerçek zamanlı PLC iletişimi

Server Name olarak KEPware.KEPServerEx.V4 seçilir. New düğmesine tıklanarak bir isim verilir.



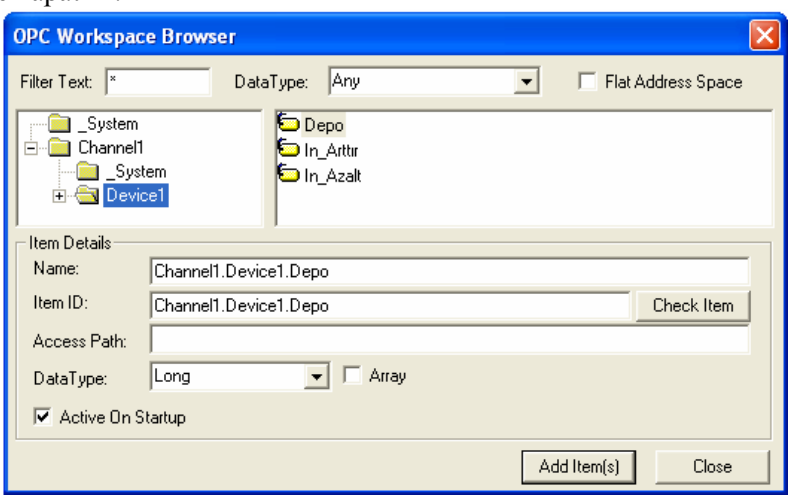
Şekil 2.73: OPC nesnesi özellikleri

OPC Groups/Items sekmesine tıklanır. Sola alanda boşluğa sağ tıklanarak bir zümre ismi girilir.



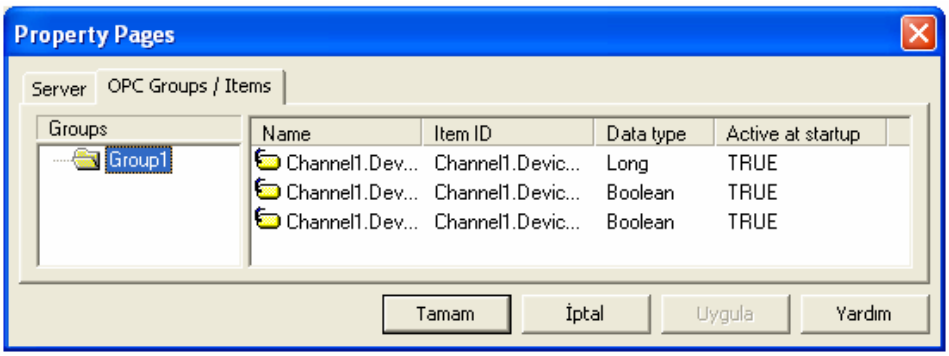
Şekil 2.74: Etiketler

Sağ alandaki boşluğa sağ tıklanarak "Add Items"'den KEPware OPC'den gelen değişkenler eklenir. Değişkenlerin her biri seçilerek Add Items düğmesine tıklanır. Daha sonra pencere kapatılır.



Şekil 2.75: Etiketler

Artık tüm değişkenler tanımlanmıştır.



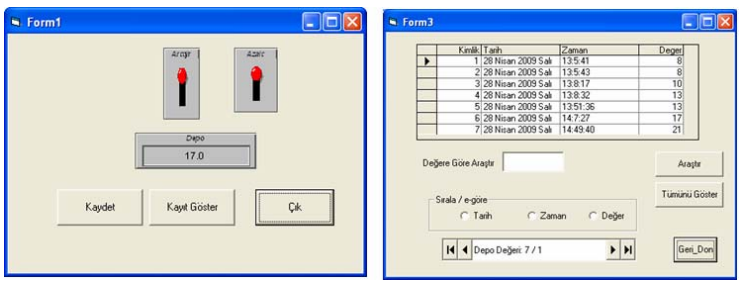
Şekil 2.76: Tanımlanan etiketler

Pencere kapatılır.

Veri kaydedilen 2.formdaki Timer'ın kodları şu şekilde değiştirilebilir.

```
Private Sub Timer1_Timer()
MyTime = Format(Time, "h:m:s")
MyStr = Format(Date, "Long Date")

                MyDate = Format(MyStr, "dddd, mmm d yyyy")

Text2.Text = MyDate
Text3.Text = MyTime

Depo_Deger = Form1.OPCComms1.Value("Group1", "Channel1.Device1.Depo")
Text4.Text = Depo_Deger

End Sub
```

Program çalıştırılır.



Şekil 2.76: Tanımlanan etiketler

UYGULAMA FAALİYETİ

Örnek 7'de verilerin kaydedilmesi için düğmeye basmak gerekir. Belirli zaman aralıklarında otomatik olarak verilerin kaydedilmesi için gereken düzenlemeyi yapınız.

Yol Gösterme:

Timer'ın Interval özelliği azami 65535'e (yak. 65 sn) ayarlanabilir. İki dakikalık periyotlarla bir olayı çalıştırılmak istenirse bu süre 120.000 milisaniyeyedir. Interval'ı 60.000'e ayarlayarak Timer'ın her bir dakikada aktif olmasını ve bu iki defa tekrarlandığında istenilen olay çalıştırılabilir.

```
Private Sub Timer1_Timer()
Static Zenta
Zenta = Zenta + 1
If Zenta Mod 2 = 0 Then

'2 dakika doldu gerekli kaydetme kodunu buraya yazınız.

End If
End Sub
```

ÖLÇME VE DEĞERLENDİRME

Aşağıdaki cümlelerde verilen bilgileri doğru veya yanlış olarak değerlendiriniz.

1. ( ) Sahadan veri toplama ve kaydetmek scadanın görevlerinden biridir.

2. ( ) "Tagname" bir değişkendir.

3. ( ) Sahanın grafiksel görünümü üzerinde veya ayrı bir ekran olarak çeşitli kontrol noktaları için, bu noktalardan gelen verilerin izlenmesi, görüntülenmesi veya raporlanması işlemi "Trends" özelliği ile sağlanır.

4. ( ) Scada sisteminin çalışması süresince sahadaki çeşitli kontrol noktalarından gelen çeşitli bilgilerin nasıl bir işleme tabi tutulacağını kullanıcı tarafından yazılan bir takım küçük program parçacıkları belirler. Bu program parçacıklarına "script"denir.

5. ( ) OPC bir arabirim **değildir.**

MODÜL DEĞERLENDİRME

PERFORMANS TESTİ

Modülde yaptığınız uygulamaları tekrar yapınız. Yaptığınız bu uygulamaları aşağıdaki tabloya göre değerlendiriniz.

AÇIKLAMA: Aşağıda listelenen ölçütleri uyguladıysanız **"Evet"** sütununa, uygulamadıysanız **"Hayır"** sütununa X işareti yazınız.		
DEĞERLENDİRME ÖLÇÜTLERİ	**Evet**	**Hayır**
1. Acces de Veri tabanı oluşturabiliyormusunuz?		
2. Veri tabanı içinde tablolar oluşturabiliyormusunuz?		
3. SCADA programı ile veri tabanına kayıt yapabiliyor musunuz ?		
4. Görsel programlama dili ile veri tabanına ulaşabiliyor musunuz ?		
5. Görsel programlama dili ile PLC'yi haberleştirebiliyor musunuz ?		
6. Görsel programlama dili ile PLC' den aldığınız verileri veri tabanına kaydedebiliyor musunuz ?		

DEĞERLENDİRME

Yapılan değerlendirme sonunda **"Hayır"** şeklindeki cevaplarınızı bir daha gözden geçiriniz. Kendinizi yeterli görmüyorsanız öğrenme faaliyetini tekrar ediniz. Bütün cevaplarınız **"Evet"** ise bir sonraki modüle geçebilirsiniz.

Öğretmeninizle iletişime geçiniz.

CEVAP ANAHTARLARI

ÖĞRENME FAALİYETİ-1'İN CEVAP ANAHTARI

1.	**D**
2.	**Birincil Anahtar**
3.	**Y**
4.	**D**
5.	**D**
6.	**A**
7.	**A**

ÖĞRENME FAALİYETİ - 2'İN CEVAP ANAHTARI

1.	**D**
2.	**D**
3.	**D**
4.	**D**
5.	**Y**

KAYNAKÇA

- NORTON Peter, **Peter Norton's Guide to Visual Basic 6**, Macmillan Computer Publishing, 1998.
- JONATHAN S., **VB Game Programming with DirectX** , Harbour by Premier Press Inc., 2002.
- Steven Holzner The Coriolis Group, **Visual Basic 6 Black Book**, 1998.
- PERRY Greg, Macmillan Computer, **Sams Teach Yourself Visual Basic 6 in 21 Days**, 1998.
- D.Bailey & E. Wright, Endüstriye **Dönük Uygulamalı: SCADA**, , Çeviren: Oktay Sunay, Bileşim Yayıncılık, 2003.
- Jeff Weigant, **Creating Human Machine Interfaces Using Visual Basic**, 2000.
- http://goforit.unk.edu
- www.kepware.com/